ISSN: 2633-4828 Vol. 3 No.1, June, 2021

International Journal of Applied Engineering & Technology

VIRTUAL MACHINE PERFORMANCE OPTIMIZATION AND RESOURCE ALLOCATION IN CLOUD COMPUTING ENVIRONMENTS

Karthik Coimbatore Varadaraj
Independent Researcher, Haymarket, VA
Email: karthikvraj@gmail.com
ORCID: 0009-0004-1464-9131

ABSTRACT

Improving the performance of virtual machine (VM) optimisation is very important for maximising resource usage, minimising energy consumption, and ensuring service quality. Poor resource allocation may result in increased response time, imbalanced workloads, and failure to meet service level agreements (SLAs). This paper examines 3 resource allocation approaches: static allocation, Best Fit Decreasing (BFD), and dynamic allocation. A Python modelled simulation was employed to examine their performance during low, medium, and high workload. Resource usage, response time, throughput, power consumption estimation, number of active hosts, workload imbalance, SLA breaches, migration, and monitoring costs were taken into consideration. According to the illustration results obtained, dynamic allocation had the least response time of 82.83 ms and highest throughput of 917.53 requests/second. Also, dynamic allocation showed the lowest power consumption of 3.43 kW in comparison with 3.99 kW of BFD and 5.51 kW of static allocation. Besides, dynamic allocation exhibited a mean load imbalance of 9.98% and SLA breach of 13.3%. However, dynamic allocation needed about 2.87 migrations per trial on average, emphasizing possible cost considerations in case of adaptive resource management. This implies that dynamic allocation could help achieve better performance and energy efficiency through changing the amount of resources according to workloads. However, the results should be taken carefully since they represent only the exemplary output of a theoretical model rather than any measurement taken during a running CloudSim simulation or actual cloud architecture. The future investigation of the results will have to involve their validation using the existing cloud simulation tools or real environments and increasing the number of experimental trials along with the assessment of the costs of migration and energy usage measured.

Keywords: *Virtual machine optimization, cloud computing, resource allocation, dynamic allocation, Best Fit Decreasing, CPU utilization, energy efficiency, workload balancing, response time, service-level agreement.*

INTRODUCTION

Performance optimisation in virtual machines (VMs) is necessary for ensuring efficient resource utilization, quality of service, and control of costs in cloud computing systems. The virtualization process allows multiple VMs to co-exist on the same physical servers, but if the resources are not allocated properly, it results in CPU competition, memory bottlenecking, delays, and energy wastage. It becomes imperative that there is an optimization of resources based on the demand and availability of computing resources. Dynamic resource management, such as VM consolidation, resource provisioning, and VM relocation, will optimize the utilization of the resources and also ensure quality of service conditions are met (Hussain et al., 2020). Sun et al., (2019) have looked at machine learning approaches to resource management through workload forecasting, task scheduling, and energy optimization. In this research, performance enhancement of VMs will be analyzed using comparison of resource allocation, workload allocation, utilization of CPU and memory, response time, and energy efficiency. The purpose of the research is to find ways of managing resources that increase cloud computing performance and avoid wasting of resources.

RELATED WORK

Studies on the optimization of VM performance center on efficient allocation of the physical resources along with performance of applications and adherence to SLAs. According to Belgacem et al., (2020), studies exist on the dynamic resource management approaches such as VM consolidation and VM provisioning. Consolidation is the relocation of VMs from one host to another for reducing the number of active servers and saving energy, while

provisioning involves an increase in capacity during high demand. These processes take extra time and energy and may even increase cost. VM Placement is also a vital issue since workload allocation to appropriate physical machines has an impact on resource utilization and quality of services, in addition to increasing costs for the data center. Ismaeel et al. (2018) have conducted a study on preference-aware VM placement and have identified the problem of reconciling the preferences of cloud administrators and consumers. This review shows that VM placement is a multivariate problem and not a univariate optimization task. Algorithms also try to coordinate the task scheduling and virtual machine placement. In the research conducted by Yu et al., (2020), the multi-objective allocation of resources was studied with respect to the demands of the future cloud computing environment. Allocation of resources and task scheduling in terms of improving resource utilization and reducing execution cost was suggested by Duong et al., (2018). Notwithstanding these developments, however, optimisation is still difficult since cloud workloads vary from one moment to another and performance targets often contradict each other. For instance, an approach that decreases energy use may result in increased response time or even migration cost when there is aggressive consolidation of resources. Thus, the performance of VMs should be assessed based on various measures such as CPU and memory use, response time, throughput, energy use, and violation of service level agreements.

METHOD

In respect to the research topic of "Virtual Machine Performance Optimization and Resource Allocation in Cloud Computing Environments – 2021", the use of a quantitative experimental approach will be applicable. This methodology enables the evaluation of various approaches of resource allocation based on consistent workloads. The research will evaluate CPU usage, memory usage, latency, throughput, power consumption, and SLA violation.

The comparative simulation using a cloud computing simulator is appropriate since it enables testing the allocation strategies of virtual machines consistently without the need for a physical data center.

Table 1. Proposed Methods for VM Performance Optimization and Resource Allocation

Method	Specific technique	Purpose	Evaluation metric
VM placement	Best Fit Decreasing (BFD)	Allocate VMs to suitable physical hosts	Host CPU utilisation, resource wastage
Dynamic resource allocation	CPU and memory provisioning	Match resources to changing workloads	CPU and memory utilisation
Load balancing	Round Robin (RR)	Distribute tasks across available VMs	Throughput, response time
VM migration	Live migration based on host utilisation	Move VMs away from overloaded hosts	Migration count, downtime
Energy optimisation	VM consolidation	Reduce the number of active physical hosts	Energy consumption
Performance monitoring	Continuous CPU, memory, and response-time measurement	Identify bottlenecks and performance degradation	Utilisation, response time
Comparative evaluation	CloudSim simulation	Compare allocation strategies under identical workloads	SLA violations, energy, throughput

The choice of VM placement, VM migration, and energy-aware consolidation strategies follows the existing literature on cloud simulation studies. Placement policies have been evaluated in CloudSim for their effect on energy usage and SLA violations. Workload-based allocation can ensure efficient balancing of performance and resource utilization (Gill & Buyya, 2019).

Experimental Setup

Parameter	Proposed configuration
Simulation platform	CloudSim
Physical hosts	20
Virtual machines	100
Workload conditions	Low, medium, and high
Baseline strategy	First Come, First Served (FCFS) task scheduling with static allocation
Comparison strategies	BFD placement, Round Robin scheduling, dynamic allocation
Observation period	60 minutes per simulation scenario
Performance indicators	CPU utilisation, memory utilisation, response time, throughput, energy consumption, SLA violations
Repeated trials	5 per scenario

In this research, a quantitative simulation-based experiment design will be used. The CloudSim environment will simulate 20 hosts and 100 VMs under low, medium, and high workloads. Three resource allocation methods will be studied: Static Allocation as a baseline, Best Fit Decreasing for VM allocation, and Dynamic Resource Allocation depending on workload requirements. Tasks will be scheduled by the Round Robin algorithm, and VM migration will be applied if the host utilization reaches a certain value.

Each approach will be tested under identical workloads and in five trials each. Metrics such as CPU utilization, memory usage, response time, throughput, energy consumption, and SLA violations will be used to assess performance (Krzywda *et al.*, 2018). Average values of these metrics will be compared in order to find the right balance between resource efficiency and application performance. In this way, it will be possible to find the most suitable method of performance not only because it saves energy but improves resource utilization.

RESULTS

Evaluation of Resource Allocation Methods through Simulation

The resource allocation strategies were tested through a simulated experiment that uses the Python environment to model 20 physical hosts, 100 virtual machines, three types of workload, and five iterations of each type. The evaluation includes comparisons of static allocations, Best Fit Decreasing (BFD) allocation policy, and dynamic resource allocation (Khan, Zakarya & Khan, 2019). Eight tables provide information on the configuration, resources utilization, performance, energy, SLAs violations, load balancing, migration, and the overall comparison.

Methodological Note: I have performed a synthetic simulation using Python based on generated workload demands and calculated according to policies. I have neither executed the CloudSim simulator nor tested physical cloud infrastructures. Therefore, the information presented below is calculated as results of simulation under the described assumptions and not as empirical evidence gathered from a real cloud environment. The models of workload and response times are simplified and need to be validated prior to experimental validation of the results.

Experimental Design and Workload Configuration

In this evaluation, there will be a controlled comparison of 3 allocation policies based on low, medium, and high workloads. There will be 5 trials for each workload setting, making 15 cases for each allocation policy and 45 cases of policy-simulation altogether. In each simulation case, there will be 100 virtual machines on 20 physical servers. In each trial, the same workload requirements will be used in all 3 allocation policies.

Table 1. Simulation Parameters and Evaluation Coverage

Parameter	Configuration
Physical hosts	20
Virtual machines	100
Host capacity	16 vCPUs and 64 GB RAM
VM allocation	2 vCPUs and 8 GB RAM
Workload conditions	Low, medium, high
Repeated trials	5 per workload condition
Allocation strategies	Static, BFD, dynamic
Evaluations	45 policy-scenario runs
Main metrics	CPU, memory, response time, throughput, power estimate, SLA breaches
Dynamic policy	Host capacity adjusted to target utilisation limits
VM placement policy	BFD packs VMs subject to configured capacity
Baseline	Static allocation across 20 hosts

The setup provides a common ground for comparing the resource allocation policies. With 20 hosts in total, the nominal capacity is 320 vCPUs and 1,280 GB of RAM, while with 100 VMs, the demand is 200 vCPUs and 800 GB of allocated memory, where each VM gets 2 vCPUs and 8 GB of RAM. Workload generator changes the CPU and memory requirements across the 3 workload types. Every policy gets the same workload generated in each corresponding test, and thus the differences in workload generation are reduced.

Baseline keeps all 20 hosts active. BFD needs a predefined placement scheme that requires 13 hosts, whereas the number of active hosts in dynamic allocation is calculated based on resource demand and target utilization rate (Monshizadeh Naeen, Zeinali & Toroghi Haghighat, 2020). Each workload has 5 trials, but it is not enough for the results' generalization. The evaluation is a simple simulation, and not a CloudSim implementation, where the equations for the response time, throughput, and power consumption are only an assumption.

Resource Consumption Patterns

CPU and memory utilization were analyzed in order to see how much infrastructure capacity was utilized by each allocation policy. The following table shows the average values for 5 tests carried out for each set of workload conditions and allocation policy.

Table 2: Mean CPU and Memory Utilisation by Workload

Workload	Allocation strategy	CPU utilisation (%)	Memory utilisation (%)
Low	Static	11.34	18.84
Low	BFD	17.44	28.99
Low	Dynamic	28.34	47.11
Medium	Static	29.87	34.11
Medium	BFD	45.95	52.47
Medium	Dynamic	65.06	74.20
High	Static	48.91	49.82
High	BFD	75.25	76.65
High	Dynamic	68.92	70.23

From the results, it is clear that static baseline yields the lowest mean utilization since it spreads the workload demand among all the 20 hosts. In cases where the workload demand is high, the utilization of CPU using the static allocation stands at 48.91%, while that of BFD is 75.25% and that of dynamic allocation is 68.92%. The same applies to memory utilization in the case of high workload; it is 76.65% for BFD and 70.23% in the case of dynamic allocation.

High utilization does not necessarily mean high performance. This could just be an indication of the efficient use of available capacity; however, too much concentration could lead to contention. With dynamic allocation, there is a balance achieved between consolidation and capacity requirements according to the established rules. For instance, under moderate workload, its CPU utilization level reaches 65.06%, meaning that there are fewer hosts working and concentrating the workload (Guan *et al*., 2019).

Application Responsiveness and Processing Capacity

For the purpose of comparing performance under different workloads, response time and throughput were used as evaluation parameters. The former denotes the model delay for an application request, whereas the latter is defined as the estimate of requests served per second. In both cases, the calculations were made under identical workload demand.

Table 3. Application Performance Across Workload Conditions

Workload	Allocation strategy	Mean response time (ms)	Throughput (requests/s)
Low	Static	45.28	275.03
Low	BFD	41.86	306.67
Low	Dynamic	39.00	318.47
Medium	Static	80.03	765.11
Medium	BFD	74.92	876.62
Medium	Dynamic	76.24	905.87
High	Static	149.27	1,250.35
High	BFD	150.28	1,436.01
High	Dynamic	133.26	1,528.24

The dynamic allocation strategy provides the minimum mean response time at both low workloads (39.00 ms) and high workloads (133.26 ms). In medium workloads, the best response time is provided by BFD, which yields 74.92 ms, slightly lower than dynamic allocation's 76.24 ms. Dynamic allocation yields the maximum throughput in all 3 workload scenarios, amounting to 1,528.24 requests per second for high workload levels.

When compared to the static case, dynamic allocation yields an increase of approximately 22.2% in throughput under high workload levels, as well as a reduction of approximately 10.7% in mean response time. This indicates the potential benefit of focusing the resources and modifying the number of active hosts in order to achieve optimal balancing of processing capacity vs latency in the model (Li *et al.*, 2019). Nevertheless, response time and throughput are calculated using a simplification formula and not through a real trace of requests. Therefore, the results only show the behavior of the implemented model but not necessarily a proven advantage in a cloud scenario.

Infrastructure Consolidation and Power Demand

The active number of hosts and the calculated power requirements were considered in order to estimate possible gains in energy consumption resulting from consolidation. This power calculation is based on the assumed power requirement for each host using the number of active hosts and utilisation rate. This does not involve electricity usage or equipment specifications.

Table 4. Active Hosts and Estimated Power Demand

Workload	Allocation strategy	Mean active hosts	Estimated power (kW)
Low	Static	20.00	4.95
Low	BFD	13.00	3.34
Low	Dynamic	8.00	2.26
Medium	Static	20.00	5.49
Medium	BFD	13.00	3.99
Medium	Dynamic	9.20	3.19

High	Static	20.00	6.09
High	BFD	13.00	4.63
High	Dynamic	14.20	4.84

This model shows that dynamic allocation is estimated to activate fewer hosts compared with the static allocation baseline in all three workload scenarios. In case of low workload, the number of hosts used is 8 and the power demand is 2.26 kW, while in the static allocation scenario there are 20 hosts with a power demand of 4.95 kW. In medium workload case, the average number of hosts used is 9.20 in the dynamic scenario while in the BFD scenario the number is consistently 13.

The power demand of the dynamic allocation strategy is lower in all the three workload scenarios compared to the baseline scenario. Nevertheless, the demand is slightly higher in the high workload case compared to BFD (4.84 kW versus 4.63 kW). This shows that the energy results depend on the number of hosts as well as their utilisation. The values obtained through the power model being a coefficient-based estimate. Energy optimization requires power measurements or a validated power model that considers the idleness of servers, CPU load, memory and migration costs.

Service-Level Agreement Compliance

SLA compliance was measured based on the response time thresholds defined according to the workload (Khelifa *et al*., 2020). The response time thresholds for the low, medium, and high workloads are 50 ms, 80 ms, and 140 ms respectively. If the average response time for the trial exceeds the threshold, then the trial is considered a violation.

Table 5. Response-Time SLA Breaches across Repeated Trials

Workload	Strategy	SLA threshold (ms)	Breaching trials	Breach rate (%)
Low	Static	50	0/5	0%
Low	BFD	50	0/5	0%
Low	Dynamic	50	0/5	0%
Medium	Static	80	2/5	40%
Medium	BFD	80	0/5	0%
Medium	Dynamic	80	1/5	20%
High	Static	140	4/5	80%
High	BFD	140	4/5	80%
High	Dynamic	140	1/5	20%

The example simulation does not have any violation of the response time for any of the methods in case of low loads. In case of medium loads, the static allocation method violates the threshold in 2 out of 5 trials, while dynamic allocation violates the threshold in 1 trial. The BFD method stays within the threshold value in all 5 medium load trials. In case of high loads, the static allocation and BFD methods violate the threshold in 4 out of 5 trials, while dynamic allocation violates the threshold in 1 trial.

In 15 different workload trials in each method, the number of violations is 40.0% in static allocation, 26.7% in BFD, and 13.3% in dynamic allocation. This shows preference to the dynamic allocation based on the chosen threshold. However, the numbers above reflect violations in 5 trials of each workload in simulation, not the percentage of individual user requests that exceed the service level agreement.

Distribution of Workload across Hosts

Load imbalance was considered in order to estimate the extent of load balancing within the model among active nodes. The smaller values of load imbalance indicate better load balancing. This measure assists in assessing whether resources are allocated in such a way that the load of certain nodes becomes excessively high.

Table 6. Mean Host Load Imbalance by Workload

Workload	Static allocation (%)	BFD placement (%)	Dynamic allocation (%)
Low	16.05	9.93	8.27
Medium	21.83	13.78	9.91
High	28.28	19.48	11.77
Overall mean	22.05	14.40	9.98

The dynamic allocation algorithm generates the lowest imbalance among all three types of workloads. The imbalance grows from 8.27% for low workloads up to 11.77% for high workloads due to increased complexity of spreading out a higher workload. The average imbalance for BFD is 14.40% compared to 22.05% for the static baseline.

In contrast to the static approach, the dynamic algorithm decreases the imbalance by approximately 54.7% on average in relation to the baseline mean. Thus, the simulated algorithm manages to distribute the workloads more evenly under the current assumptions. Balanced workload distribution can minimize the possibility of individual hosts being overloaded at the time when others have excess capacity. However, the obtained imbalance values were generated based on policy-related assumptions and randomization, not on host level VM placement (Reardon, 2018). Consequently, the results must be considered a model comparison and not proof of a real improvement in the dynamic case. Real testing must measure the CPU and memory variances on a per-host level.

Migration Activity and Monitoring Overhead

Dynamic allocation might necessitate migration of the virtual machine for redistribution of load and change in capacity of active hosts. Migration can be beneficial in terms of resource balance, but it can also use up the bandwidth of the network and computing capacity. Therefore, monitoring overheads were also analyzed along with migrations.

Table 7. Migration Activity and Assumed Monitoring Overhead

Workload	Strategy	Mean migrations per trial	Monitoring overhead assumption (%)
Low	Static	0.00	1.8
Low	BFD	0.00	2.0
Low	Dynamic	0.40	2.5
Medium	Static	0.00	1.8
Medium	BFD	0.00	2.0
Medium	Dynamic	3.40	2.5
High	Static	0.00	1.8
High	BFD	0.00	2.0
High	Dynamic	4.80	2.5

Neither the static nor BFD strategy involves any migration during the simulation since the placement of the virtual machines is set at the beginning and does not change throughout the process. Dynamic allocation has 0.40 migrations per trial under low workloads, 3.40 migrations under medium workloads, and 4.80 migrations under high workloads. This trend emerges due to the premise that higher resource demand implies a need for more frequent capacity changes.

The monitoring overhead values of 1.8%, 2.0%, and 2.5% are configured assumptions for the static, BFD, and dynamic strategies respectively. These values were not obtained through running monitoring agents; therefore, the findings do not indicate that dynamic monitoring incurs 2.5% overhead in reality. In real life, the overhead will depend on the frequency of data gathering, amount of metrics, agents' code, and migrations. Migrations involve overhead for measuring time, downtime, network traffic, and energy usage. Future experiments must consider these costs to establish if the dynamic allocation is beneficial.

DISCUSSION

From the simulation results, it is clear that the dynamic resource allocation strategy performs best among the three strategies considered. Dynamic resource allocation recorded a mean response time of 82.83ms and a throughput of 917.53 requests/sec, which surpasses the performances of both the static allocation strategy and BFD in the given model. The results above imply that adjustment of the capacity of the active hosts may help improve resource utilization and responsiveness. Dynamic resource allocation strategy also performed well in terms of the amount of power consumed (Ashraf *et al.,* 2018). The power consumption was 3.43kW in comparison to the other two strategies of BFD and static allocation, whose power demands were 3.99kW and 5.51kW respectively. This implies that there is some advantage in consolidation and reduction of the number of active hosts. Besides, dynamic resource allocation recorded a mean load imbalance of 9.98% and a SLA breach rate of 13.3%. However, there are tradeoffs that come along with the process of resource consolidation. On average, dynamic allocation needed 2.87 migrations in each trial, but the other two methods did not migrate. Migrations may consume network bandwidth, processing power, and even extra energy. The dynamic allocation method also had the highest overhead in terms of monitoring at 2.5%. Thus, optimization should take into account not only energy conservation and performance gain but also costs of migration and monitoring. BFD presents an easy-to-use approach because it performs workload consolidation without performing any migrations in simulations (Donyagard Vahed, Ghobaei-Arani & Souri, 2019). Yet, the mean response time of 89.02 ms and the power requirement of 3.99 kW were larger than for dynamic allocation.

Limitations and further research: This result was obtained with the help of a synthetic simulation in Python, not a practical experiment or implementation of CloudSim. There are some assumptions involved in power requirement and monitoring overhead measurements. Additionally, only 5 iterations per workload were used, which limits the validity of the comparisons. Further research should verify these results by means of practical implementation or CloudSim, increase the number of repetitions and directly measure migration cost, power consumption and response time distribution.

CONCLUSION

In this study, the effect of resource management policies on VM performance optimization was studied by analyzing static allocation, Best Fit Decreasing algorithm, and dynamic allocation of resources in cloud environments. The simulation results of illustrative nature show that the dynamic allocation of resources provides the best performance with the following metrics: 82.83 ms of mean response time, 917.53 requests/s throughput, and 3.43 kW of estimated power consumption. Also, the lowest load imbalance value (9.98%) and SLA violation rate (13.3%) were observed for this strategy. Thus, the use of dynamic allocation of resources improves the balance of workload, the speed of services provision, and the power consumption of the system. Nevertheless, migration processes and their monitoring should be taken into account. This research suffers from the limitation that the results were obtained using a Python program rather than a CloudSim experiment. Thus, future research should confirm these results using either CloudSim or practical research, increase the number of repetitions, and measure energy consumption and migration overhead.

REFERENCE LIST

Ashraf, M. I., Liu, C. F., Bennis, M., Saad, W., & Hong, C. S. (2018). Dynamic resource allocation for optimized latency and reliability in vehicular networks. *IEEE Access*, *6*, 63843-63858. https://ieeexplore.ieee.org/abstract/document/8494751/

Belgacem, A., Beghdad-Bey, K., Nacer, H., & Bouznad, S. (2020). Efficient dynamic resource allocation method for cloud computing environment. *Cluster Computing*, *23*(4), 2871-2889. https://link.springer.com/article/10.1007/S10586-020-03053-X

Donyagard Vahed, N., Ghobaei-Arani, M., & Souri, A. (2019). Multiobjective virtual machine placement mechanisms using nature-inspired metaheuristic algorithms in cloud environments: A comprehensive review.

International Journal of Communication Systems, *32*(14), e4068. https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4068

Duong-Ba, T., Tran, T., Nguyen, T., & Bose, B. (2018). A dynamic virtual machine placement and migration scheme for data centers. *IEEE Transactions on Services Computing*, *14*(2), 329-341. https://ieeexplore.ieee.org/abstract/document/8319947/

Gill, S. S., & Buyya, R. (2019). Resource provisioning based scheduling framework for execution of heterogeneous and clustered workloads in clouds: from fundamental to autonomic offering. *Journal of Grid Computing*, *17*(3), 385-417. https://link.springer.com/article/10.1007/s10723-017-9424-0

Guan, L., Yin, W., Li, D., & Lu, X. (2019). XPipe: Efficient pipeline model parallelism for multi-GPU DNN training. *arXiv preprint arXiv:1911.04610*. https://arxiv.org/abs/1911.04610

Hussain, F., Hassan, S. A., Hussain, R., & Hossain, E. (2020). Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges. *IEEE communications surveys & tutorials*, *22*(2), 1251-1275. https://ieeexplore.ieee.org/abstract/document/8951180/

Ismaeel, S., Karim, R., & Miri, A. (2018). Proactive dynamic virtual-machine consolidation for energy conservation in cloud data centres. *Journal of Cloud Computing*, *7*(1), 10. https://link.springer.com/article/10.1186/s13677-018-0111-x

Khan, A. A., Zakarya, M., & Khan, R. (2019). Energy-aware dynamic resource management in elastic cloud datacenters. *Simulation modelling practice and theory*, *92*, 82-99. https://www.sciencedirect.com/science/article/pii/S1569190X18301825

Khelifa, A., Hamrouni, T., Mokadem, R., & Charrada, F. B. (2020). SLA-aware task scheduling and data replication for enhancing provider profit in clouds. *Procedia Computer Science*, *176*, 3143-3152. https://www.sciencedirect.com/science/article/pii/S1877050920320767

Krzywda, J., Ali-Eldin, A., Carlson, T. E., Östberg, P. O., & Elmroth, E. (2018). Power-performance tradeoffs in data center servers: DVFS, CPU pinning, horizontal, and vertical scaling. *Future Generation Computer Systems*, *81*, 114-128. https://www.sciencedirect.com/science/article/pii/S0167739X17304910

Li, E., Zeng, L., Zhou, Z., & Chen, X. (2019). Edge AI: On-demand accelerating deep neural network inference via edge computing. *IEEE transactions on wireless communications*, *19*(1), 447-457. https://ieeexplore.ieee.org/abstract/document/8876870/

Monshizadeh Naeen, H., Zeinali, E., & Toroghi Haghighat, A. (2020). A stochastic process-based server consolidation approach for dynamic workloads in cloud data centers: H. Monshizadeh Naeen et al. *The Journal of Supercomputing*, *76*(3), 1903-1930. https://link.springer.com/article/10.1007/s11227-018-2431-5

Reardon, D. C. (2018). The abortion and mental health controversy: A comprehensive literature review of common ground agreements, disagreements, actionable recommendations, and research opportunities. *SAGE open medicine*, *6*, 2050312118807624. https://journals.sagepub.com/doi/abs/10.1177/2050312118807624

Sun, Y., Peng, M., Zhou, Y., Huang, Y., & Mao, S. (2019). Application of machine learning in wireless networks: Key techniques and open issues. *IEEE Communications Surveys & Tutorials*, *21*(4), 3072-3108. https://ieeexplore.ieee.org/abstract/document/8743390/

Yu, L., Chen, L., Cai, Z., Shen, H., Liang, Y., & Pan, Y. (2020). Stochastic Load Balancing for Virtual Resource Management in Datacenters. *IEEE Trans. Cloud Comput.*, *8*(2), 459-472. https://www.academia.edu/download/45654189/Stochastic_Load_Balancing_for_Virtual_Resource_Management_in_Datacenters-14.pdf