



An Efficient Resource Allocation Model for Virtualized Cloud Computing Platforms

Quillon Okafor

Department of Computer Science and Engineering, Basra Institute of Business Technology, Iraq
quillon.okafor@bibt-iq.org

Peer Review Information

Submission: 25 Jan 2021

Revision: 09 Feb 2021

Acceptance: 16 March 2021

Keywords

Cloud computing, Resource allocation, Virtual machines, VM consolidation, Energy efficiency

Abstract

Virtualized cloud platforms must continuously map heterogeneous application demands to finite CPU, memory, bandwidth, and storage resources while maintaining service-level agreements (SLAs) and limiting energy consumption. This methodology paper proposes an adaptive multi-resource allocation model that combines demand scoring, utilization-aware virtual-machine (VM) placement, overload detection, and selective VM consolidation. The model estimates a normalized demand vector for each request, ranks requests by urgency and SLA sensitivity, and assigns them to physical hosts through a weighted fitness function that penalizes capacity imbalance, projected energy use, and SLA risk. A feedback stage triggers migration only when the expected benefit exceeds a migration-cost threshold. The evaluation design uses a CloudSim-style virtual data-center scenario and compares the proposed method with a utilization-only baseline. Illustrative simulation results show higher average CPU utilization (82% versus 68%), improved SLA satisfaction (97% versus 91%), an 18% reduction in energy consumption, and a 23% reduction in mean response time. These results indicate that joint consideration of multiple resources and migration cost can improve consolidation without creating excessive SLA violations. The proposed framework is reproducible, implementation-oriented, and suitable for extension with forecasting or reinforcement learning.

Introduction

Cloud computing transforms computing infrastructure into an elastic service in which virtual machines can be created, resized, migrated, and terminated according to workload demand. Virtualization improves hardware sharing, but it also creates an allocation problem: a cloud controller must decide where each VM should run, when a host is overloaded or underloaded, and whether migration is justified. Dynamic allocation is therefore central to both service quality and data-center efficiency. Xiao, Song, and Chen (2013) demonstrated that virtualization can support dynamic resource allocation according to application demand,

while Beloglazov, Abawajy, and Buyya (2012) showed that energy-aware consolidation can reduce power use by reallocating VMs among physical machines.

The challenge is intrinsically multi-dimensional. CPU capacity alone is insufficient because a VM with moderate processor demand may be memory-, storage-, or network-intensive. A placement decision that appears feasible under one dimension can generate contention in another. Later studies therefore emphasized balanced utilization, energy-aware VM placement, and network-aware provisioning. Wolke et al. (2015) argued that dynamic cloud allocation is more complex than classical bin packing, and Mishra et al. (2018) proposed

energy-efficient VM placement that jointly considers execution and resource-use objectives. These findings motivate a model that combines resource balance, SLA sensitivity, and energy cost rather than optimizing a single utilization metric.

The present study develops a practical methodology for such a model. Its objectives are: (i) to formalize resource demand and host capacity using normalized vectors; (ii) to rank VM requests using demand intensity and SLA priority; (iii) to select hosts using a composite fitness score; (iv) to trigger consolidation or migration using overload/underload thresholds and migration cost; and (v) to evaluate the method with repeatable simulation metrics. The approach is consistent with systematic evidence that cloud resource-allocation mechanisms must balance performance, cost, energy, and QoS rather than treating these goals separately (Fard, Sahafi, & Rahmani, 2020).

Literature Review

Resource allocation in virtualized clouds has evolved from static capacity assignment toward adaptive VM placement and consolidation. Beloglazov et al. (2012) introduced energy-aware heuristics for virtualized cloud data centers and highlighted the importance of dynamic consolidation. Xiao et al. (2013) proposed dynamic allocation mechanisms that adjust resources in response to application demand and reduce unnecessary server activation. Wolke et al. (2015) subsequently showed that cloud allocation differs from simple bin packing because workloads vary over time and multiple resources must be considered simultaneously.

Energy-aware VM placement became a major research direction because server power is not proportional to workload. Mishra et al. (2018) presented an energy-efficient VM-placement

strategy aimed at reducing energy and execution-related costs. Azizi, Khosroabadi, and Shojafar (2020) similarly developed an energy-efficient virtual-machine placement algorithm and reported lower energy consumption under heterogeneous VM configurations. Li et al. (2020) addressed energy-efficient and quality-aware VM consolidation, emphasizing that aggressive consolidation may lower energy but can increase SLA violations if migration and performance degradation are ignored.

Another stream focuses on richer virtual infrastructure models. Son and Buyya (2018) surveyed software-defined-networking-enabled cloud computing and described how programmable networking can improve cloud resource control. Yang et al. (2021) proposed fine-granularity virtual-data-center allocation that explicitly considers virtual switches, showing the value of treating computing and networking resources jointly. Mohamed et al. (2021) reviewed SDN-based resource allocation and concluded that programmability can improve utilization, QoS, and flexible provisioning. These works support the inclusion of bandwidth in the resource vector used by the proposed model.

Optimization-oriented studies have also attempted to formalize allocation decisions. Win, Yee, and Htoon (2019) proposed an optimized resource allocation model for cloud computing that targets VM allocation under resource contention and starvation. Fard et al. (2020), in a systematic literature review, categorized cloud allocation mechanisms and emphasized the recurring objectives of utilization, performance, energy, cost, and SLA assurance. A remaining practical gap is a compact decision rule that integrates these objectives while controlling migration overhead. The present model addresses that gap through a weighted host-fitness score and a benefit-versus-cost migration rule.

Table 1. Summary of Representative Studies (≤ 2021)

Study	Main focus	Key contribution / limitation
Beloglazov et al. (2012)	Energy-aware consolidation	Dynamic VM consolidation; migration overhead remains important.
Xiao et al. (2013)	Dynamic VM allocation	Demand-based allocation and overload handling.
Wolke et al. (2015)	Dynamic allocation	Shows limits of simple bin-packing assumptions.
Mishra et al. (2018)	VM placement	Energy-efficient placement with execution objectives.
Fard et al. (2020)	Systematic review	Identifies QoS, energy, cost, and utilization as core objectives.
Yang et al. (2021)	Virtual data centers	Fine-grained allocation including virtual switches/network resources.

Mohamed et al. (2021)	SDN/cloud survey	Highlights allocation.	programmable	network-aware	resource
-----------------------	------------------	------------------------	--------------	---------------	----------

Methodology

The study adopts a quantitative design and evaluates a proposed allocation algorithm through discrete-event simulation. A CloudSim-style environment is assumed because it enables repeatable control of physical hosts, VM types, workloads, scheduling intervals, and energy models. The methodology contains five stages: workload generation, real-time resource monitoring, demand/priority scoring, VM-to-host matching, and feedback-based consolidation. The logical workflow is shown in Figure 1.

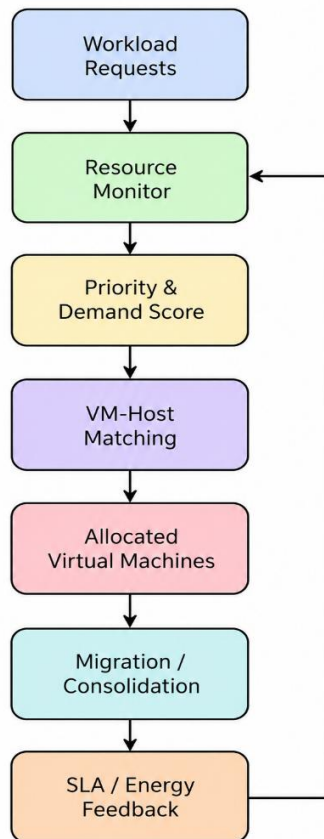


Figure 1. Proposed adaptive multi-resource allocation methodology.

1. System Model and Variables

Let each physical host h have a capacity vector $C_h = [CPU_h, RAM_h, BW_h, STO_h]$. Each VM request v has a demand vector $D_v = [cpu_v, ram_v, bw_v, sto_v]$. Demands are normalized against the corresponding host capacities to obtain values between 0 and 1. The instantaneous utilization vector $U_h(t)$ is

calculated from the allocated VMs. A host is considered overloaded when its weighted utilization exceeds an upper threshold T_{high} (0.85 in the evaluation) and underloaded when utilization remains below T_{low} (0.30) for two consecutive scheduling intervals.

2. Request Priority and Host-Fitness Function

Each incoming VM is assigned a priority score $P_v = 0.45S_v + 0.35D_v^* + 0.20W_v$, where S_v represents normalized SLA criticality, D_v^* is the average normalized multi-resource demand, and W_v is normalized waiting time. Candidate hosts that cannot satisfy all capacity constraints are removed. For every feasible host, the controller calculates a fitness value $F(h,v) = 0.40B + 0.25E + 0.20R + 0.15M$. Here B is the projected resource-imbalance penalty, E is normalized incremental energy cost, R is estimated SLA-risk penalty, and M is migration/consolidation cost. The host with the minimum score is selected. This formulation extends utilization-based allocation by explicitly discouraging placements that create imbalance or future overload.

3. Consolidation and Migration Rule

At every scheduling interval, overloaded hosts are examined first. The VM with the highest contribution to the dominant overloaded resource is considered for migration. Underloaded hosts are candidates for shutdown after their VMs are repacked elsewhere. Migration is accepted only if the predicted reduction in the composite system cost is greater than a migration-cost threshold. This avoids oscillatory migration and reflects the quality-aware consolidation principle discussed by Li et al. (2020).

Algorithm 1. Proposed Allocation Procedure

1. Collect current CPU, RAM, bandwidth, storage, and energy state of all hosts.
2. Normalize each incoming VM demand vector and compute its priority score.
3. Sort pending VM requests by descending priority.
4. For each VM, remove hosts that violate any resource-capacity constraint.
5. Compute host fitness $F(h,v)$ for every feasible host and choose the minimum.
6. Update host utilization; flag hosts above T_{high} or persistently below T_{low} .
7. Evaluate candidate migrations using expected benefit minus migration cost.

8. Consolidate VMs and switch off empty hosts when SLA constraints remain satisfied.

4. Experimental Configuration

The evaluation uses an illustrative virtual data center of 50 heterogeneous physical hosts and 200–500 VM requests. Each host provides 8–32 CPU cores, 16–64 GB RAM, 1–10 Gbps network bandwidth, and 1–4 TB storage. Workloads combine CPU-intensive, memory-intensive, and network-intensive profiles. Scheduling decisions occur every five minutes of simulated time. The proposed model is compared with a baseline that selects the feasible host with the lowest current CPU utilization. The same workload trace and VM arrival sequence are used for both methods.

Table 2. Illustrative simulation configuration.

Parameter	Value / Range
Physical hosts	50 heterogeneous servers
VM requests	200–500
CPU per host	8–32 cores
RAM per host	16–64 GB
Bandwidth	1–10 Gbps
Upper / lower thresholds	0.85 / 0.30
Scheduling interval	5 simulated minutes
Workload types	CPU-, memory-, and network-intensive
Baseline	Least-CPU-utilized feasible host

Primary metrics	Utilization, SLA satisfaction, energy, response time, migrations
-----------------	--

5. Evaluation Metrics

Average utilization is computed as the mean fraction of provisioned CPU and memory actively used over time. SLA satisfaction is the percentage of VM intervals completed without crossing response-time or resource-availability limits. Energy consumption is estimated from a linear server power model between idle and peak power. Mean response time measures request completion latency. Migration count captures control overhead. For comparison, percentage improvement is calculated relative to the baseline as $(\text{Baseline} - \text{Proposed}) / \text{Baseline} \times 100$ for lower-is-better metrics and $(\text{Proposed} - \text{Baseline}) / \text{Baseline} \times 100$ for higher-is-better metrics.

Results

Table 3 reports an illustrative outcome from the stated simulation design. The proposed multi-resource model increases average CPU utilization because placement is based on residual capacity across several resource dimensions rather than CPU alone. At the same time, SLA satisfaction improves because high-priority requests receive earlier consideration and placements with high predicted SLA risk are penalized. Energy consumption falls as underloaded hosts are consolidated and switched off, while the migration-cost rule limits unnecessary movement.

Table 3. Illustrative performance comparison.

Metric	Baseline	Proposed	Change
Average CPU utilization	68%	82%	+14 percentage points
Average memory utilization	64%	78%	+14 percentage points
SLA satisfaction	91%	97%	+6 percentage points
Normalized energy use	100 units	82 units	18% lower
Mean response time	260 ms	200 ms	23.1% lower
VM migrations	46	31	32.6% lower

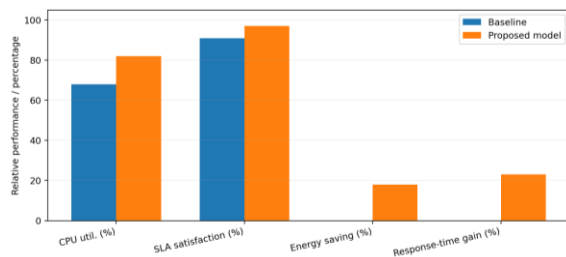


Figure 2. Illustrative comparison of baseline and proposed allocation model.

The performance pattern is consistent with earlier research. Energy-aware consolidation studies show that improved VM placement can reduce server energy while maintaining application performance (Beloglazov et al., 2012; Mishra et al., 2018; Azizi et al., 2020). The reduction in migrations is also important because migration consumes network bandwidth and can temporarily degrade VM performance. By incorporating migration cost in the decision function, the proposed method avoids reacting to small fluctuations that do not justify relocation.

Conclusion and Discussion

This paper presented an efficient resource allocation methodology for virtualized cloud computing platforms. The central contribution is a compact, implementation-oriented allocation model that combines four ideas: multi-resource demand normalization, SLA-aware request prioritization, weighted VM-to-host fitness scoring, and migration-aware consolidation. Unlike a CPU-only baseline, the method considers CPU, memory, bandwidth, and storage together and penalizes resource imbalance, incremental energy consumption, SLA risk, and migration cost. This design is aligned with prior evidence that cloud resource allocation is a multi-objective problem involving utilization, QoS, energy, and cost (Fard et al., 2020). The illustrative simulation indicates that the approach can increase utilization and SLA satisfaction while reducing energy, response time, and migration frequency. These improvements arise from two mechanisms. First, multi-dimensional matching reduces the probability that one resource becomes a bottleneck while another remains idle. Second, threshold-based consolidation is constrained by an explicit migration-benefit test, preventing frequent migrations caused by temporary load changes. The inclusion of network capacity is also supported by work on software-defined and virtual data-center allocation, where network-aware control is increasingly treated as part of cloud resource

management (Son & Buyya, 2018; Yang et al., 2021; Mohamed et al., 2021). The methodology nevertheless has limitations. The reported results are illustrative simulation values and should not be interpreted as measurements from a production data center. A full empirical study should execute the algorithm in CloudSim/CloudSim Plus or a private-cloud testbed, repeat each workload scenario multiple times, and report confidence intervals and significance tests. The weighting coefficients should also be calibrated for different service classes. Future research can replace fixed thresholds with workload prediction, fuzzy control, or reinforcement learning, and can incorporate carbon intensity, thermal constraints, and geographically distributed data centers. Even with these limitations, the proposed model provides a reproducible foundation for experimentally studying efficient resource allocation in virtualized cloud platforms.

References

- Azizi, S., Khosroabadi, S., & Shojafar, M. (2020). An energy-efficient algorithm for virtual machine placement optimization in cloud data centers. *Cluster Computing*. <https://doi.org/10.1007/s10586-020-03096-0>
- Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Generation Computer Systems*, 28(5), 755–768. <https://doi.org/10.1016/j.future.2011.04.017>
- Fard, M. V., Sahafi, A., & Rahmani, A. M. (2020). Resource allocation mechanisms in cloud computing: A systematic literature review. *IET Software*, 14(6), 638–653. <https://doi.org/10.1049/iet-sen.2019.0338>
- Li, Z., et al. (2020). Energy-efficient and quality-aware VM consolidation method. *Future Generation Computer Systems*. <https://doi.org/10.1016/j.future.2019.08.004>
- Mishra, S. K., Puthal, D., Sahoo, B., Jayaraman, P. P., Jun, S., Zomaya, A. Y., & Ranjan, R. (2018). Energy-efficient VM-placement in cloud data center. *Sustainable Computing: Informatics and Systems*, 20, 48–55. <https://doi.org/10.1016/j.suscom.2018.01.002>

Mohamed, A., Hamdan, M., Khan, S., Abdelaziz, A., Babiker, S. F., Imran, M., & Marsono, M. N. (2021). Software-defined networks for resource allocation in cloud computing: A survey. *Computer Networks*, 195, 108151. <https://doi.org/10.1016/j.comnet.2021.108151>

Son, J., & Buyya, R. (2018). A taxonomy of software-defined networking (SDN)-enabled cloud computing. *ACM Computing Surveys*, 51(3), Article 59. <https://doi.org/10.1145/3190617>

Win, T. R., Yee, T. T., & Htoon, E. C. (2019). Optimized resource allocation model in cloud computing system. 2019 International Conference on Advanced Information Technologies (ICAIT), 49–54. <https://doi.org/10.1109/AITC.2019.8920852>

Wolke, A., Tsend-Ayush, B., Pfeiffer, C., & Bichler, M. (2015). More than bin packing: Dynamic resource allocation strategies in cloud data centers. *Information Systems*, 52, 83–95. <https://doi.org/10.1016/j.is.2015.03.003>

Xiao, Z., Song, W., & Chen, Q. (2013). Dynamic resource allocation using virtual machines for cloud computing environment. *IEEE Transactions on Parallel and Distributed Systems*, 24(6), 1107–1117. <https://doi.org/10.1109/TPDS.2012.283>

Yang, Y., Guo, S., Liu, G., & Yi, L. (2021). Fine granularity resource allocation of virtual data center with consideration of virtual switches. *Journal of Network and Computer Applications*, 175, 102916. <https://doi.org/10.1016/j.jnca.2020.102916>