STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LXX**, Number 1-2, 2025
DOI: 10.24193/subbi.2025.05

# THE G1 GAUSSIAN-TYPE RESOURCE ALLOCATION POLICY FOR VIRTUALIZED DATA CENTERS: THE SCALING PROBLEM AND VARIATION OF PARAMETERS

CORA CRĂCIUN

Abstract. Virtualized data centers use techniques such as resource consolidation to reduce the energy consumption and load balancing methods to improve performance. A previously proposed resource allocation policy of Gaussian type, named G1, has a behaviour in-between resource consolidation and load balancing. This policy was previously evaluated by simulation in a small data center configuration. This paper evaluates G1 for higher dimensional data centers with up to 400 hosts and 800 virtual machines, in order to establish if the policy scales with the dimension of the data center. G1 is compared with the First Fit heuristic by simulation for time-varying workloads. Metrics such as energy consumption, the mean number of active hosts, and the number of VMs migrations are calculated for different parametrizations of the score function of the G1 policy.

## 1. Introduction

Data centers use virtualization [1] as a way to manage the resources efficiently in dynamic conditions with time-varying workloads. The jobs are encapsulated in virtual machines (VMs) which are deployed on the physical resources and may be reconfigured or relocated. Through consolidation of the virtual machines on the hosts, the energy consumption of the data centers decreases. Heuristics such as First Fit (FF) and Best Fit (BF) proved efficient for workload consolidation [2, 3, 4, 5]. A drawback of the consolidation methods is that the virtual machines are packed too tightly on the physical

Received by the editors: 21 July 2025.

2010 *Mathematics Subject Classification.* 68M14, 68M20.

1998 *CR Categories and Descriptors.* C.2.4 [**Computer-Communication Networks**]: Distributed Systems – *Network operating systems*; C.4 [**Computer Systems Organization**]: Performance of Systems – *Design studies*.

*Key words and phrases.* G1 Gaussian-type policy, resource allocation, energy consumption, virtual machine migration.

hosts and therefore when the workload increases some virtual machines from the overloaded hosts should be migrated to other physical machines with spare resources [6, 7]. Virtual machine migrations, however, come with energy and performance overheads. The physical machines involved in the migration process consume more energy, the performance of the jobs encapsulated in the virtual machines decreases, and the network may become overused [8, 9, 10]. Unlike consolidation methods, load balancing techniques [11] aim to improve the performance of the data centers. The virtual machines are distributed on more hosts and afford better workload increases. When the workload is low, however, the physical machines may become underused and again the virtual machines are migrated and the empty hosts are switched off. The consolidation and load balancing techniques were combined with proportional sharing and Dynamic Voltage and Frequency Scaling procedures in order to address the performance and energy consumption problems [6, 12].

As a way to deal with the energy consumption - performance trade-off problem, we proposed previously two Gaussian-type resource allocation policies, G1 and G2, which pack the VMs on hosts less tightly than the greedy heuristics such as First Fit and Best Fit but more tightly than the load balancing procedures [13]. In the resource allocation process, the two policies maximize score functions of Gaussian shape and having some adjustable parameters. The G1 policy may be used in online conditions and chooses the hosts with the highest scores for the enqueued VMs. The $G_1$ score function depends on the used resources of the physical machines. The G2 policy, on the other hand, is more suitable for offline conditions and chooses the (VM,host) pairs that maximize a $G_2$ score function depending on the required resources by the VM and the available resources of the host. The single constraining resource for the VMs' allocation to the hosts was the CPU, but the Gaussian policies may be extended to other type of resources such as memory and disk space. The Gaussian-type policies were tested by simulation in a framework [14] built on the Haizea lease scheduler [15, 17, 18, 19, 20], for configurations with few resources, namely one with insufficient physical resources, (8 hosts, 40 VMs), and one with sufficient physical resources, (20 hosts, 40 VMs). The VMs had time-varying workloads, randomly generated in a given range, and were migrated when the hosts on which were deployed became overloaded. Metrics such as energy consumption, mean number of active hosts, VM migration number, flow time, and makespan were computed. The simulations showed that the G1 and G2 policies used more hosts than the FF and BF heuristics and their decreasing forms and thus consumed more energy, but reduced the number of virtual machines migrations, affording better the workload variations [13, 14, 16]. It should be noted that the reduction of the number of VMs migrations was mainly achieved by a less tightly packing of the VMs with the Gaussian-type resource allocation policies. The migration of the VMs was performed with a VM migration policy, such as migrating the VMs with the least or highest CPU requirements.

This paper investigates the G1 policy in more detail. Using a new software implementation of the data center in Python, the G1 policy is tested for data centers with up to 400 hosts and 800 VMs. The energy consumption, the mean number of active hosts, and the VM migration number are computed for each data center configuration. Moreover, for a configuration with 300 hosts and 600 VMs, the simulations are performed for different parametrizations of the $G_1$ score function.

The paper is structured as follows. Next section reviews some investigations which aimed to reduce the number of VMs migrations in data centers. The section Computational details gives the expression of the $G_1$ score function and contains details about the software implementation. The section Results and discussion reports the computed metrics for the data centers of different sizes and for the G1 policy with various parametrizations. The final section concludes the current work.

## 2. Related work

Beside reducing the energy consumption or improving performance several studies tried to reduce the number of VMs migrations. In one of the algorithms used by the pMapper controller [21], Verma et al. utilized an incremental First Fit Decreasing resource allocation method combined with a VM migration policy that migrates the VMs with the smallest size, thus minimizing the power usage and reducing the VMs migrations. In [22], Ferreto et al. applied linear programming or heuristics for dynamic consolidation of the VMs and a migration controlled method which prevented to migrate the VMs with steady requirements. By performing in this way, the VMs migrations were reduced, with a small overhead in the number of used physical machines. Sandpiper [23] used algorithms to find the hotspots in data centers and to remove them by VMs operations such as VM migration, swapping, or resizing. The migration overhead was minimized by migrating the VMs with the highest volume-to-size ratio, from the physical machines with the highest volume to the least used physical machines. In order to reduce the energy consumption, the number of VMs migrations, and the SLA violations in heterogeneous data centers, Beloglazov et al. [5] combined a Power Aware Best Fit Decreasing VM packing method with a host overloading detection procedure based on thresholds and a Minimization of Migrations policy. The simulation experiments were performed using the CloudSim framework [24, 25]. In [26], Hermenier et al. proposed a constraint programming procedure called BtrPlace in order to consolidate the VMs on the physical machines. When several constraints were not satisfied, a reconfiguration plan was generated, which enhanced with different strategies aimed to reduce the VMs relocations. Mastroianni et al. [27, 28] proposed a decentralized resource management method for Cloud data centers, called ecoCloud, in order to minimize the number of used physical machines, but without decreasing performance. The algorithm mapping the new or relocated VMs on servers solved a CPU and RAM-constraint Bin Packing Problem. The destination hosts were chosen probabilistically by performing Bernoulli trials [27]. Our Gaussian-type score functions resemble the probability functions used by ecoCloud for VM mapping and migration. Ashraf and Porres [29] proposed a multi-objective algorithm based on an ant colony system, aiming to reduce the number of used physical machines and the number of VMs migrations.

## 3. Computational details

A data center has a number $N_H$ of homogeneous hosts and $N_V$ virtual machines which must be mapped on physical machines for their deployment, based on their CPU requests. The data center has sufficient hosts for running all VMs. All initial VMs start processing simultaneously. Each VM has a trace of CPU requests lasting 40 minutes. The CPU requests are uniformly distributed numbers in the $[10, 40]$ range (in percents), rounded to the closest integer value. The CPU request of a VM changes at each 2 minutes. The VMs are mapped on hosts based on the G1 resource allocation policy or, for comparison, the FF heuristic. In time, because the VMs' workload changes, the hosts may become overused. In such a case, selected VMs are migrated to other hosts.

### 3.1. **VM scheduling.**

The VMs are enqueued and are assigned to the hosts in their enqueuing order. When some VMs are suspended from the overloaded hosts, they are immediately assigned to other hosts which have spare resources. The time delays due to VMs' suspension, migration, and resumption on the new hosts are ignored.

### 3.2. **Resource allocation.**

In current work, the G1 resource allocation policy is compared with the FF heuristic, known for its efficiency in reducing the energy consumption. The G1 policy uses a $G_1(U_{CPU})$ score function in order to map the VMs on hosts, where $U_{CPU}$ is the CPU usage of a physical machine. The VMs are assigned in order to those physical machines which maximize the score function. A VM may be assigned to a host if its CPU requirements,

$R_{CPU}$, are less than the host's available resources, $A_{CPU}$. The total CPU capacity of a host, $T_{CPU} = U_{CPU} + A_{CPU}$, is 100%.

The $G_1(U_{CPU})$ function is the following Gaussian:

$$G_1(U_{CPU}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(U_{CPU}-\mu)^2}{2\sigma^2}\right],$$

where

$$\mu = \frac{Thr_L + Thr_H}{2}, \quad \sigma = \frac{Thr_H - Thr_L}{2\sqrt{2}\,\mathrm{erf}^{-1}(a)}.$$

$\mathrm{erf}^{-1}$ is the inverse error function.

The mean value, $\mu$, of the two threshold values, $Thr_L$ and $Thr_H$, is the middle point of the Gaussian. The threshold values fall in the [0, $T_{CPU}$] interval and $Thr_H > Thr_L$. The parameter $a$ represents the area between the threshold values, below the Gaussian, and belongs to the $(0, 1)$ interval.

3.3. **VMs suspension and migration.** A host is overloaded when $U_{CPU} > T_{CPU}$. The first trial when a host becomes overloaded is to suspend a single VM, with the smallest possible CPU request. If this is not possible in order to remove the overload then more VMs are suspended. The VMs of the overloaded host are sorted in increasing order of CPU requirements and the VMs are suspended in order until the overload is cleared. This process may lead to unnecessary suspensions. Then, the list of suspended VMs is scanned in decreasing order after CPU request and the VMs which still do not cause the overload are restored. This kind of suspension was used in reference [14]. The reason for suspending the VMs with smaller CPU requests is to reduce the overhead of their migration and to find easily spare resources on the hosts. The hosts are checked for overloading in the increasing order of their identifier. The VMs suspended from all overloaded hosts are sorted in increasing order of the initial host identifiers (the initial hosts are the hosts from which the VMs are migrated) and then after the VMs' identifier. Then, all simultaneously suspended VMs are mapped on other hosts with the resource allocation policy, with no time delay.

In current work only the overused hosts were subject to VMs suspensions. It is known that when the hosts are underused, all their VMs may be migrated and consolidated on other used hosts. The previously underused hosts, now freed, are switched off in order to reduce the energy consumption.

3.4. **The computed metrics.** The metrics computed which enabled us to compare the FF and G1 policies are the CPU energy consumption of the hosts, the mean number of active hosts, and the number of VMs migrations. The additional energy consumption of the hosts involved in migrations was neglected.

The CPU energy consumption may be computed as the following sum:

$$E = \sum_s P(\tau_s)(\tau_{s+1} - \tau_s),$$

where $P(\tau_s)$ is the power used by all physical machines at the scheduling time $\tau_s$. The scheduling times were considered the times when the VMs' CPU requirements changed, namely 0, 2, ..., 38 minutes. For each host, the power was considered linearly dependent on the hosts' CPU utilization, $U_{CPU}$, according to the relation [30]:

$$P = P_{idle} + (P_{busy} - P_{idle})U_{CPU}.$$

$P_{idle}$ is the power used when no VM is deployed on the host, while $P_{busy}$ is the power when the host is fully utilized. The values used in computation were those from reference [5], $P_{busy} = 250$ W and $P_{idle} = 70\% P_{busy} = 175$ W. The energy consumption was computed in kilowatt hour.

The mean number of active hosts was computed by summing up the number of used hosts at all scheduling times and by dividing the result with the number of scheduling times. The number of VMs migrations is the total number of migrations experienced by the VMs in the makespan.

3.5. **Software design.** The software used in this work is a simplified implementation of the Haizea-based framework presented in [14], which may be used for higher number of VMs and hosts. Current software is not based on Haizea [15] or other available toolkit. The scheduling, resource allocation, and VM migration policies implemented in the software are identical with some of the policies from the Haizea-based framework. The reason for using a simplified implementation of the Haizea-based framework was to enable comparison of the current work with the results from [13, 16].

The CPU trace of each VM is implemented as a list containing the CPU requests of the workload, at each 2 minutes, for a total makespan of 40 minutes. The CPU requests are randomly generated between 10% and 40% as in references [13, 14, 16], again for direct comparison of present results with those from previous work. The VMs and hosts are implemented as Python classes. The VM class defines the VM identifier (ID), the state of the VM (not assigned to a host, assigned to a host, suspended for further migration), the current physical machine hosting the VM, the next host to which the VM is migrated, the CPU trace list with the workload requirements. The Host class defines the host identifier, the total, available, and used CPU of the host, the state of the host (switched off, busy, overloaded), the idle power usage, the total power usage, a list with power usage at each scheduling time, a list with the IDs of the current VMs deployed on the host, a list with the IDs of the

VMs suspended from the host, a function for updating the resources of the host.

The software creates a list with $N_V$ VM objects and a list with $N_H$ Host objects. Since the data center has enough physical machines, all VMs are mapped on hosts at the initial time, with the chosen resource allocation policy (FF, G1), and the hosts' resources are updated. At the other scheduling times, coinciding with the times of changing the VMs' CPU requests, the first operation is detection of overloaded hosts. Then one or more VMs are suspended from the overloaded hosts and the hosts' CPU resources are updated accordingly. The VMs to be migrated are mapped on other hosts with enough available resources, using the chosen resource allocation policy, and again the hosts' resources are updated. The scheduling, resource allocation, and VM migration policies used by the software are those described in the previous sections. At each scheduling time, the hosts' power usage and the number of used hosts are computed. After the processing of the VM CPU traces, the energy consumption, the mean number of active hosts, and the number of VMs migrations are reported.

## 4. Results and discussion

4.1. **The scaling problem.** In order to investigate if the G1 resource allocation policy scales with the dimension of the data center, different simulation experiments were performed for the following configurations: (50 hosts, 100 VMs), (100 hosts, 200 VMs), (200 hosts, 400 VMs), (300 hosts, 600 VMs), and (400 hosts, 800 VMs). The number of hosts was chosen such that to have enough physical resources for VMs with CPU requirements in the [10,40] range. 20 simulation experiments, with different VM CPU traces, were performed for each data center configuration. Each experiment was performed for the FF heuristic and G1 policy, with the same VM CPU trace. The parameters for the $G_1$ score function were $Thr_L = 40$, $Thr_H = 80$, and $a = 0.8$, as in references [13, 14, 16]. Figure 1 presents the results of the 20 simulation experiments, in boxplot representation, for the three metrics computed in this paper, while Table 1 contains the relative contrasts of the mean values of the metrics for G1 compared with FF. A positive value in the table indicates that the metric is higher for G1 than for FF and a negative value corresponds to a lower value for G1 than for FF. As can be seen in Figure 1 the number of used hosts is significantly smaller than that provisioned, for both policies. The energy consumption when using the G1 policy is higher than for FF in the limit of 5%, for all data center configurations. This additional energy consumption for G1 comes from using a higher number of hosts. The VM migration number, on the other hand, is significantly lower (in the limit of 25%) for G1

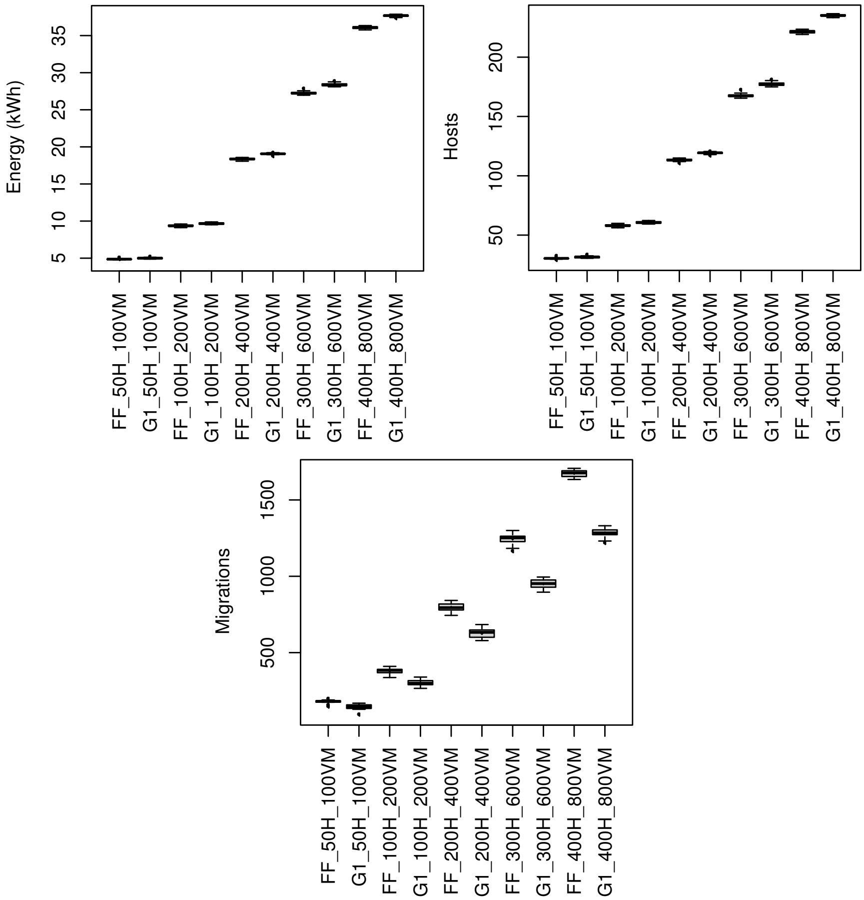


FIGURE 1. Boxplot representation [31] of the consumed energy, the mean number of active hosts, and the VM migration number for the G1 and FF policies, in data centers with different number of hosts and VMs. Simulation conditions: 40 min long VM CPU traces, 20 simulation experiments.

compared to FF. The relative values for the energy, mean number of active hosts, and VM migration number in absolute value increase slightly with the dimension of the data center. G1 reduces the number of VMs migrations with an energy consumption overhead, packing the VMs on hosts less tightly than the FF heuristic.

TABLE 1. The energy consumption, the mean number of active hosts, and the VM migration number relative contrasts (in percents), calculated from the mean values on 20 simulation experiments, for the indicated G1_H_VM policies, with the parameters $a = 0.8$, $\mu = 60$, and $\Delta Thr = 40$, with respect to FF_H_VM policies

| Compared policies | $\theta^{E}_{rel}$ | $\theta^{H}_{rel}$ | $\theta^{NM}_{rel}$ |
|---|---|---|---|
| G1_50H_100VM vs. FF_50H_100VM | 2.5 | 3.5 | -19.1 |
| G1_100H_200VM vs. FF_100H_200VM | 3.2 | 4.4 | -20.6 |
| G1_200H_400VM vs. FF_200H_400VM | 3.9 | 5.4 | -21.8 |
| G1_300H_600VM vs. FF_300H_600VM | 4.1 | 5.8 | -23.7 |
| G1_400H_800VM vs. FF_400H_800VM | 4.4 | 6.2 | -23.4 |

4.2. **Variation of parameters for the $G_1$ score function.** For the (300 hosts, 600 VMs) data center configuration, simulation experiments were performed for different parametrizations of the $G_1$ score function. The Gaussian function was centered at the values $\mu = 40$, 50, 60, and 70, the distance between the thresholds was $\Delta Thr = 40$, and the $a$ parameter had the values 0.4, 0.6, 0.8, and 0.95. Figure 2 presents the energy, the mean number of active hosts, and the VM migration number metrics for 20 experiments with the FF and G1_$a$_$\mu$ policies. Table 2 shows the relative contrasts of the mean values of the metrics for G1 with respect to FF. For energy the relative contrasts are in the limit of 7.5%, for the mean number of active hosts the relative contrasts are in the limit of 10%, while for the number of migrations the relative contrasts are in the limit of 38%. The $a$ parameter changes the metrics only slightly. For instance the relative contrast for energy for G1_0.4_40 versus FF is 7%, while for G1_0.95_40 versus FF is 6.8%. For the number of migrations, the relative contrast for G1_0.4_40 versus FF is -37.0% and for G1_0.95_40 versus FF is -37.3%. The $\mu$ parameter and thus the values of the two thresholds, on the other hand, change significantly the relative contrasts of the metrics. When the $\mu$ value increases, the relative contrasts of G1 versus FF decrease, so G1 resembles more the FF heuristic. For instance, the relative contrast of the energy for G1_0.6_40 versus FF is 7.1%, but only 1.7% for G1_0.6_70 versus FF. The relative contrasts for the VM migration number for the two cases are -37.6% and -12.8%, respectively. It should be noted that at a VM migration number relative contrast of about -12%, the energy overhead of G1 is below 2%.

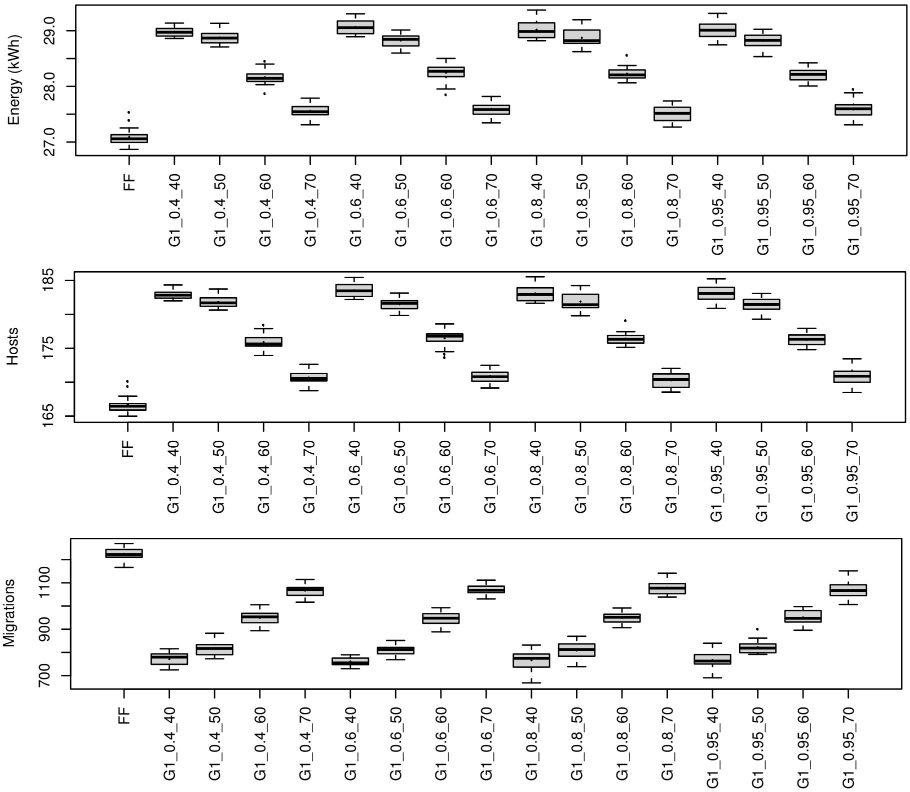


FIGURE 2. Boxplot representation [31] of the energy consumption, the mean number of active hosts, and the VM migration number for FF and G1_$a$_$\mu$ policies with different parameters, in data centers with 300 hosts and 600 VMs. Simulation conditions: 40 min long VM CPU traces, 20 simulation experiments.

## 5. Conclusions

This paper investigated in more detail a previously defined Gaussian-type resource allocation policy for virtualized data centers, named G1. The policy was compared with the First Fit heuristic by simulation, from the point of view of energy consumption, mean number of used active hosts, and number of VMs migrations performed. Simulation experiments with time-varying workloads were performed for different dimensions of the virtualized data center. The G1 policy scales with the dimension of the data center, showing the same trend of reducing significantly the number of VMs migrations, with a moderate

TABLE 2. The energy consumption, the mean number of active hosts, and the VM migration number relative contrasts (in percents), calculated from the mean values on 20 simulation experiments, for the indicated G1_$a$_$\mu$ policies with respect to the First Fit policy, in a data center with 300 hosts and 600 VMs

| Policy | $\theta_{rel}^{E}$ | $\theta_{rel}^{H}$ | $\theta_{rel}^{NM}$ |
|---|---|---|---|
| G1_0.4_40 | 7.0 | 9.7 | -37.0 |
| G1_0.4_50 | 6.3 | 8.8 | -33.5 |
| G1_0.4_60 | 4.1 | 5.7 | -23.5 |
| G1_0.4_70 | 1.8 | 2.5 | -13.5 |
| G1_0.6_40 | 7.1 | 9.9 | -37.6 |
| G1_0.6_50 | 6.4 | 9.0 | -34.1 |
| G1_0.6_60 | 4.0 | 5.6 | -22.5 |
| G1_0.6_70 | 1.7 | 2.4 | -12.8 |
| G1_0.8_40 | 6.9 | 9.6 | -37.6 |
| G1_0.8_50 | 6.4 | 9.0 | -34.5 |
| G1_0.8_60 | 4.1 | 5.7 | -22.3 |
| G1_0.8_70 | 1.7 | 2.3 | -13.0 |
| G1_0.95_40 | 6.8 | 9.5 | -37.3 |
| G1_0.95_50 | 6.3 | 8.7 | -33.6 |
| G1_0.95_60 | 4.1 | 5.7 | -22.0 |
| G1_0.95_70 | 1.5 | 2.1 | -11.9 |

energy consumption overhead. In the resource allocation process, the G1 policy maximizes a Gaussian score function depending on three parameters, two thresholds (or the mean of these thresholds and the distance between them) and an area parameter. Simulations with different parametrizations of the score function showed that the mean value of the thresholds affects the values of the metrics (energy, number of hosts, number of VMs migrations) significantly, while the area parameter has a negligible effect.

## REFERENCES


[1] Gerald J Popek and Robert P Goldberg. Formal requirements for virtualizable third generation architectures. *Communications of the ACM*, 17(7):412–421, 1974.

[2] Laura Grit, David Irwin, Aydan Yumerefendi, and Jeff Chase. Virtual machine hosting for networked clusters: building the foundations for "autonomic" orchestration. In *Proceedings of the First International Workshop on Virtualization Technology in Distributed Computing (VTDC)*, pp. 7, 2006.

[3] Michael Cardosa, Madhukar R Korupolu, and Aameek Singh. Shares and utilities based power consolidation in virtualized server environments. In *Proceedings of the 11th*

*IFIP/IEEE International Conference on Symposium on Integrated Network Management (IM)*, pp. 327–334, 2009.

[4] Mark Stillwell, Frédéric Vivien, and Henri Casanova. Virtual machine resource allocation for service hosting on heterogeneous distributed platforms. In *Proceedings of the 2012 IEEE 26th International Parallel & Distributed Processing Symposium (IPDPS)*, pp. 786–797, 2012.

[5] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5):755–768, 2012.

[6] Anton Beloglazov, Rajkumar Buyya, Young C Lee, and Albert Zomaya. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *Advances in Computers*, 82:47–111, 2011.

[7] Anton Beloglazov and Rajkumar Buyya. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. *Concurrency and Computation: Practice & Experience*, 24(13):1397–1420, 2012.

[8] William Voorsluys, James Broberg, Srikumar Venugopal, and Rajkumar Buyya. Cost of virtual machine live migration in Clouds: a performance evaluation. In *Proceedings of the 1st International Conference on Cloud Computing (CloudCom)*, pp. 254–265, 2009.

[9] Anja Strunk and Waltenegus Dargie. Does live migration of virtual machines cost energy?. In *Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pp. 514–521, 2013.

[10] Jinho Hwang, Sai Zeng, Frederick y Wu, and Timothy Wood. A component-based performance comparison of four hypervisors. In *Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 269–276, 2013.

[11] Minxian Xu, Wenhong Tian, and Rajkumar Buyya. A survey on load balancing algorithms for virtual machines placement in cloud computing. *Concurrency and Computation: Practice & Experience*, 29(12):e4123, 2017.

[12] Kyong H Kim, Anton Beloglazov, and Rajkumar Buyya. Power-aware provisioning of virtual machines for real-time Cloud services. *Concurrency and Computation: Practice & Experience*, 23(13):1491–1505, 2011.

[13] Cora Crăciun and Ioan Salomie. Gaussian-type resource allocation policies for virtualized data centers. *Studia Universitatis Babeş-Bolyai, Informatica*, LXI(2):94–109, 2016.

[14] Cora Crăciun and Ioan Salomie. A filter-based dynamic resource management framework for virtualized data centers. *Studia Universitatis Babeş-Bolyai, Informatica*, LXII(1):32–48, 2017.

[15] Haizea. `http://haizea.cs.uchicago.edu/`.

[16] Cora Crăciun and Ioan Salomie. Bayesian analysis of resource allocation policies in data centers in terms of virtual machine migrations. In *Proceedings of 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 511–518, 2017.

[17] Borja Sotomayor, Kate Keahey, and Ian Foster. Combining batch execution and leasing using virtual machines. In *Proceedings of the 17th International Symposium on High Performance Distributed Computing (HPDC)*, pp. 87–96, 2008.

[18] Borja Sotomayor, Rubén S Montero, Ignacio M Llorente, and Ian Foster. An open source solution for virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing, Special Issue on Cloud Computing*, 2009.

[19] Borja Sotomayor, Rubén S Montero, Ignacio M Llorente, and Ian Foster. Resource leasing and the art of suspending virtual machines. In *Proceedings of the 11th IEEE International Conference on High Performance Computing and Communications (HPCC)*, pp. 59–68, 2009.

[20] Borja Sotomayor Basilio. Provisioning computational resources using virtual machines and leases. PhD Dissertation, University of Chicago, Illinois, USA, 2010.

[21] Akshat Verma, Puneet Ahuja, and Anindya Neogi. pMapper: power and migration cost aware application placement in virtualized systems. In *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware (Middleware)*, pp. 243–264, 2008.

[22] Tiago C Ferreto, Marco AS Netto, Rodrigo N Calheiros, and César AF De Rose. Server consolidation with migration control for virtualized data centers. *Future Generation Computer Systems*, 27:1027–1034, 2011.

[23] Timothy Wood, Prashant Shenoy, Arun Venkataramani, and Mazin Yousif. Sandpiper: Black-box and gray-box resource management for virtual machines. *Computer Networks*, 53(17):2923–2938, 2009.

[24] Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Journal of Software: Practice and Experience*, 41(1):23–50, 2011.

[25] CloudSim. `http://www.cloudbus.org/cloudsim/`.

[26] Fabien Hermenier, Julia Lawall, and Gilles Muller. BtrPlace: A flexible consolidation manager for highly available applications. *IEEE Transactions on Dependable and Secure Computing*, 10(5):273–286, 2013.

[27] Carlo Mastroianni, Michela Meo, and Giuseppe Papuzzo. Probabilistic consolidation of virtual machines in self-organizing cloud data centers. *IEEE Transactions on Cloud Computing*, 1(2):215–228, 2013.

[28] Carlo Mastroianni, Michela Meo, and Giuseppe Papuzzo. Self-economy in cloud data centers: Statistical assignment and migration of virtual machines. In *Proceedings of the 17th International Conference on Parallel Processing (Euro-Par)*, pp. 407–418, 2011.

[29] Adnan Ashraf and Ivan Porres. Multi-objective dynamic virtual machine consolidation in the cloud using ant colony system. *International Journal of Parallel, Emergent and Distributed Systems*, 33(1):103–120, 2018.

[30] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz A Barroso. Power provisioning for a warehouse-sized computer. In *Proceedings of the 34th annual International Symposium on Computer architecture (ISCA)*, pp. 13–23, 2007.

[31] The R Project for Statistical Computing. `https://www.r-project.org/`.

Faculty of Physics, Babeş-Bolyai University, Cluj-Napoca, Romania
*Email address*: `cora.craciun@ubbcluj.ro`