An Energy-SLA Aware VM Selection Algorithm for Dynamic Virtual Machine Consolidation

Pankaj Jain[1], Sanjay Kumar Sharma[2]
[1,2] Department of Computer Science, Banasthali Vidyapith, Niwai, Rajasthan, India

Abstract: - Energy efficiency is an essential issue in cloud computing. It has a massive financial and environmental effect that needs attention. A good VM-host mapping achieves low energy consumption while minimizing the number of migrations and Service Level Agreement Violation (SLAV). Dynamic Virtual Machine Consolidation (DVMC) is an excellent solution to reduce Energy Consumption (EC). However, an aggressive DVMC may increase the SLAV. So there is a need to balance the trade-off between EC and SLAV. Therefore, This work presents a Knapsack-based VM Selection (KVMS) algorithm. It works on the dynamic 0/1 knapsack approach and selects the VM with the maximum ratio of its CPU utilization and migration time. Simulation results prove that proposed KVMS reduces SLAV and Energy-SLAV (ESV) at most 60% and 64%, respectively.

***Keywords**: Cloud Computing, Virtual Machine Consolidation, Energy Consumption, Service Level Agreement Violation, VM Selection.*

1. Introduction

Cloud computing has recently arisen as a reliable and trusted computing technology that enhances the utilization of virtualized resources and services for end users [1]. It provides software and hardware as computing resources through the internet by the pay-as-you-use concept [2]-[4]. Figure 1 shows a cloud computing environment consisting of web resources like servers, applications, storage, and software platforms [5].

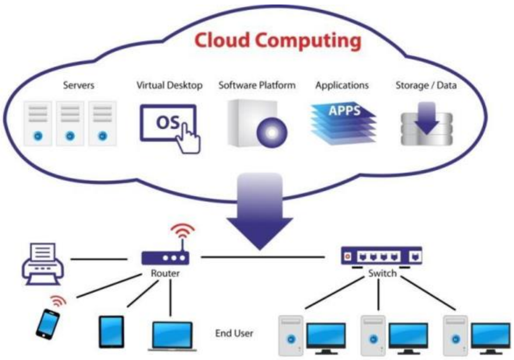

Figure 1: Cloud computing environment [4]

From 2010 to 2018, cloud data center computing instances and workload grew by six times which impact the operation cost and environmental pollution both [6],[7]. It increases energy consumption. So, there is a need for energy-aware resource allocation in cloud data center. This can be done by an efficient mapping of Virtual Machine (VM) to Physical Machine (PM).

The Service Level Agreement (SLA) serves as a legal contract between the Cloud Service Provider (CSP) and Cloud User (CU), guaranteeing that the CU will receive uninterrupted service quality. In the event of an SLA violation (SLAV), the CSP will face financial penalties [8]-[10]. To prevent such violations, the CSP must minimize EC while maintaining SLAV. However, there exists a a tradeoff between minimizing EC and avoiding SLAV. In cloud computing, reducing EC can involve consolidating Virtual Machines (VMs) onto fewer hosts, which could potentially degrade performance and increase the risk of SLAV. Therefore, finding a harmonious equilibrium between EC and SLAV is crucial. Dynamic Virtual Machine Consolidation (DVMC) is an essential step for resource allocation [11]-[13] in datacenter.

Virtual Machine Consolidation (VMC) is an effective solution that helps to optimize resource utilization. DVMC consists of four phases as below [14][15] :

1. To detect underloaded host: A host underloaded detection algorithm detects whether a host is underloaded or not.
2. Overloaded Host detection. A host underloaded detection algorithm detects whether a host is overloaded or not.
3. Selection of VMs from overloaded Hosts for migration: After detecting the overloaded host, a VM selection algorithm selects VMs to be migrated to another host.
4. Appropriate Placement of selected VM : An appropriate host is selected to placed migrated Vms from overloaded/underloaded host.

This paper focuses on third step of DVMC. The process of VM selection for migration plays a vital role in the domain of energy-aware cloud computing. A study report shows that an ideal server consumes approximately 70% of its peak power utilization [16], and most servers operate only from 10% to 50% of their maximum capacity, leading to low server utilization [17]. Many idle or underloaded servers are the major contributor to energy wastage. Thus minimization of the active host and load balancing are efficient approaches to reduce EC in CDC. Therefore VM migration plays a vital role in DVMC and load balancing [18]. However, too many VM migrations also adversely affect the efficiency of the host and data center, as it is a resource-intensive technique that continuously demands CPU, memory, and communication bandwidth[19].

A lot of work [20],[21] has been proposed to improve DVMC. The efficiency of DVMC can be enhanced by improving the host underloaded/overloaded detection algorithm, VM selection policy, and VM placement policies. Literature [22] proposed Markov Power aware Best Fit Decreasing (MPBFD) for VM placement. Host Utilization Aware (HUA) algorithm is proposed in [23]. In [24], Multi-objective Dynamic VM Consolidation (MDVMC) is proposed to improve VMP. Existing work focused on one parameter either EC or SLAV. Host overloaded and VM selection algorithms are proposed in [25],[26]. Literature [27] proposed minimum migration time over deviation for VM selection. Vms are selected for migration based on minimum data transfer rate in [28]. Literature [23] selects Vm that have highest unsatisfied resource requirements. VM Migration Overhead Algorithm is used in [24] for Vm selection. Literatute [29] selects Vms of maximum CPU or memory utilization. This work presents a Knapsack-based VM Selection (KVMS) algorithm. It works on the dynamic 0/1 knapsack approach and selects the VM with the maximum ratio of its CPU utilization and migration time.

2. METHOD

2.1 Proposed Knapsack-based VM Selection (KVMS) algorithm for DVMC

A VM Selection policy finds the VMs that should be migrated from the overloaded host. Choosing the wrong VM may result in an increased number of migrations, leading to higher costs and energy consumption for the data center. Therefore, an effective VM selection policy is necessary for DVMC. This section contains a knapsack [30] based VM selection algorithm.

Knapsack is a dynamic computer programming approach and operates on the principle of a thief who possesses a bag with a capacity of C (kg) and must choose and fill it with the available items to obtain the highest possible profit. The decision is taken based on n items having weights and profit values. There are two types of knapsack problems; one is a 0/1 knapsack in which items cannot be broken, and the second one is a fractional knapsack in which items can be selected infractions. This work uses the 0/1 knapsack technique because the VMs cannot be allocated infractions. Therefore, the proposed algorithm is called the Knapsack-based VM Selection (KVMS) algorithm. In KVMS, CPU utilization represents bag capacity; VMs in an overloaded host represents the items; CPU utilization of VM represents the item's weight, and VM's migration time represents the profit earned. The proposed work selects the VM with the maximum ratio of its CPU utilization and migration time. That means that instead of selecting VM with minimum CPU utilization [31], KVMS finds the VM with greater utilization and shorter VM migration time so that a VM that consumes more CPU capacity and can be migrated in less time is selected for migration. As a result, the proposed algorithm reduces the number of VM migrations and energy consumption.

A VMx is selected for migration if it satisfies Equation 1:

$$\left(\frac{\text{Utilization}}{\text{MigrationTime}}\right)_x > \left(\frac{\text{Utilization}}{\text{MigrationTime}}\right)_y \tag{1}$$

Here, y all other VMs in migration List

The Utilization factor does not depend solely on CPU utilization (UCPU); therefore, Utilized Bandwidth (UBW) and Utilized RAM (URAM) are used to evaluate the Utilization of VM on the host. It can be calculated using equation 2:

$$\text{Utilization} = U_{CPU} * U_{BW} * U_{RAM} \tag{2}$$

2.2 Proposed Algorithm

Algorithm 1 shows the steps of KVMS algorithm. KVMS utilizes the concept of knapsack for selecting VMs from overloaded hosts. It chooses a Vm for migration with a greater utilization and migration time ratio. Here, MigratableVmsList is the list of VMs in overloaded host that participate in migration process. If this list is empty, then algorithm returns null in step 3. Steps 6 to 10 are repeated for each VMx. Line 6 checks if VMx is already in migration then continue the process for next VM. Step 8 calculates Utilization by using equation 2 and steps 9 & 10 find VM that has greater (Utilzation/MigrationTime) than other VMs. Step 12 return that selected VM.

Algorithm 1: Knapsack-based VM Selection (KVMS) algorithm

Input: host, MigratableVmsList

Output: selected VM

1. Start
2. if MigratableVmsList is empty then
3.   return null
4. Initialize VM← null
5. for each VMx (x ∈ MigratableVmsList)
6.   if isInMigration (VMx) is true then
7.   Continue Step 5
8.   Calculate Utilization using equation 2
9.   if VMx satisfies condition given in equation 1 then
10.   Assign VM← VMx
11. end for
12. return VM
13. End

3. Results And Discussion

3.1 Experimental setup

Performance of proposed algorithms is evaluated using CloudSim. Total 800 heterogeneous hosts are taken. Here, real dataset Planet Lab that is furnished as a part of the CoMon project is used as workload. It is accessible from Beloglazov's GitHub repository (https://github.com/beloglazov/planetlab-workload-traces). The workload traces of one day (20110303) are taken to conduct the experiment. Simulation limit is taken 24 hour for experiments.

3.2 Performance analysis and results

The performance of proposed work is measured with the help of performance metrics i.e., Energy Consumption (EC), SLAV, and Energy and SLA Violation (ESV). The simulation is conducted for all combinations of the existing four host overload detection (HOD) algorithms (Iqr, Lr, Mad, Thr), four VM selection (VMS) algorithms (Mc, Mmt, Mu, Rs), and one VM placement algorithm (PABFD). For comparison, proposed KVMS is combined with the above four HOD and PABFD. These combinations are compared based on performance metrics.

Table 1 shows the result of simulation and maximum improvement from using KVMS. Best results are shown as "Bold". Figures 1 to 3 show graphical representation of performance metrics. Figure 1 shows the comparative analysis of energy consumption. it can be seen that among existing algorithms, LrKVMS gives best results for EC. KVMC improves EC atmost 19%, 18%, 20%, and 19% with Iqr, Lr, Mad, and Thr, respectively. Figure 2 represents the results of SLAV. It can be seen from figure that IqrKVMS gives minimum SLAV. KVMC minimizes SLAV atmost 60%, 42%, 53%, and 54% with Iqr, Lr, Mad, and Thr, respectively. . Figure 3 represents the results of ESV. It can be seen from figure that IqrKVMS gives minimum ESV. KVMC minimizes ESV atmost 64%, 45%, 61%, and 58% with Iqr, Lr, Mad, and Thr, respectively.

Table 1: Comparisons of KVMS with existing VMS algorithms

Performance Metrics	Algorithms (VMS ↓ / HOD →)	Iqr	Lr	Mad	Thr
EC	**KVMS**	163	**142**	160	167
	Mc	178	150	176	183
	Mmt	188	163	184	191
	Mu	202	174	200	206
	Rs	180	149	191	184
	Maximum Improvement(%)	19	18	20	19
SLAV*10-2 (%)	**KVMS**	**0.289**	0.403	0.348	0.331
	Mc	0.726	0.677	0.739	0.697
	Mmt	0.303	0.463	0.331	0.324
	Mu	0.472	0.592	0.510	0.481
	Rs	0.695	0.694	0.743	0.716
	Maximum Improvement(%)	60	42	53	54
ESV*10-2 (%)	**KVMS**	**47.0**	57.3	55.7	55.3
	Mc	129.2	101.6	130.1	127.6
	Mmt	57.0	75.5	60.9	61.9
	Mu	95.3	103.0	102.0	99.1
	Rs	123.7	103.4	141.9	131.7
	Maximum Improvement (%)	64	45	61	58

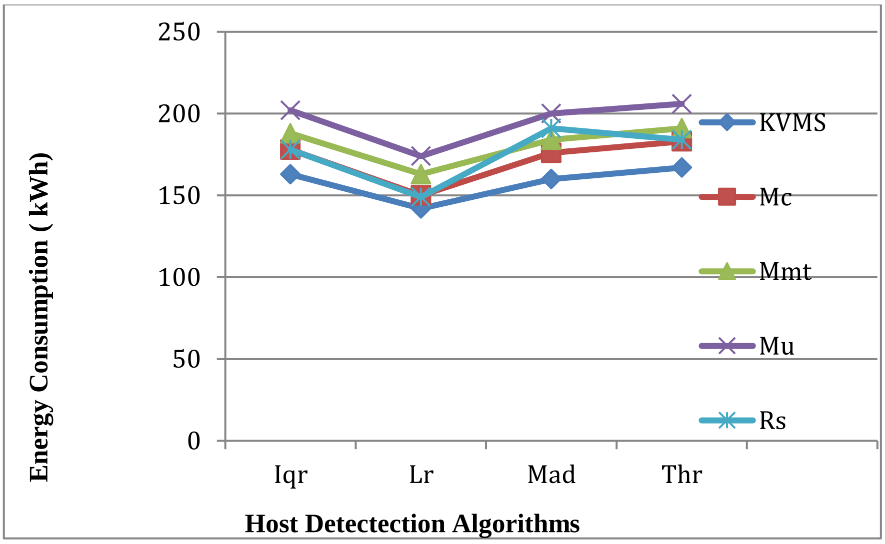

Figure 1: Energy Consumption for proposed and existing algorithms

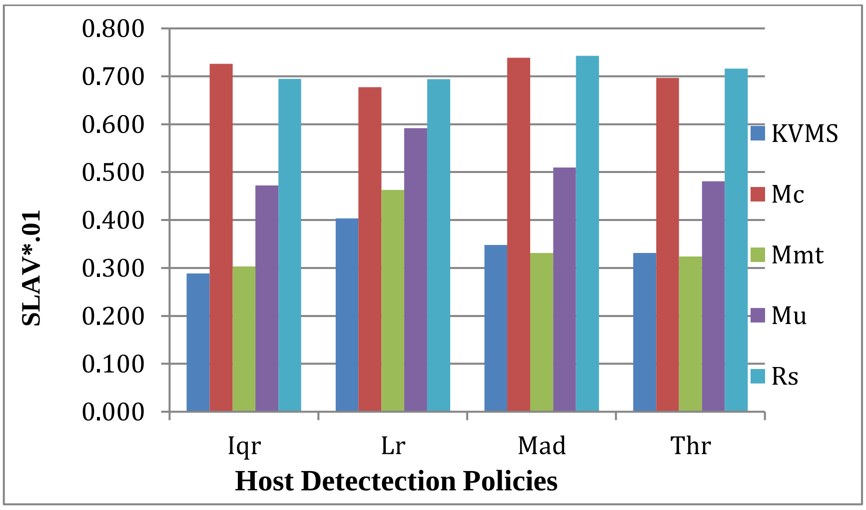

Figure 2: SLAV for proposed and existing algorithms

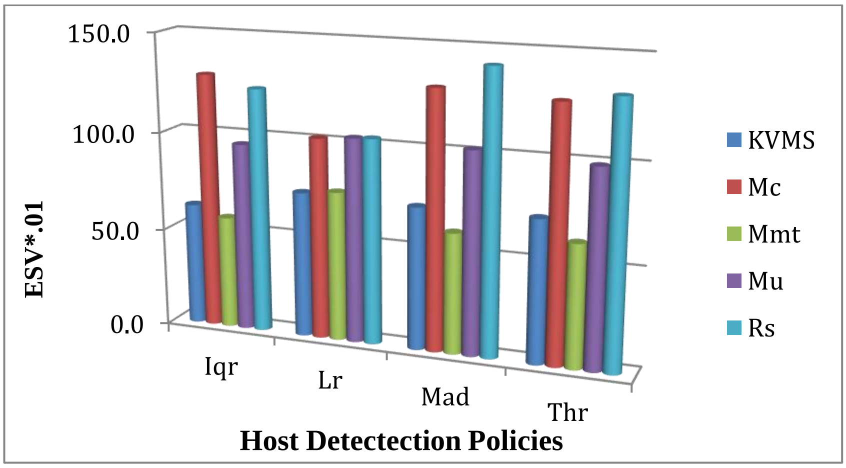

Figure 3: ESV for proposed and existing algorithms

4. Conclusion

This paper presents Knapsack-based VM Selection (KVMS) algorithm for DVMC. KVMS utilizes the concept of knapsack for selecting VMs from overloaded hosts. It chooses a Vm for migration with a greater utilization and migration time ratio. Simulation results proved that using KVMS for VMS improves performance metrics, i.e., EC, SLAV, and ESV. Simulation results prove that proposed KVMS reduces SLAV and ESV at most 60% and 64%, respectively.

References

[1] P. Mell and T. Grance, “The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology Special Publication 800-145,” Sep. 2011, doi: http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf

[2] J. Daniels, “Server virtualization architecture and implementation,” Crossroads, vol. 16, no. 1, pp. 8–12, Sep. 2009, doi: https://doi.org/10.1145/1618588.1618592.

[3] B. Speitkamp and M. Bichler, “A Mathematical Programming Approach for Server Consolidation Problems in Virtualized Data Centers,” IEEE Transactions on Services Computing, vol. 3, no. 4, pp. 266–278, 2010, doi: https://doi.org/10.1109/TSC.2010.25.

[4] Gill, S.S., Buyya, R., Chana, I., Singh, M., Abraham, " BULLET:particle swarm optimization based scheduling technique for provisioned cloud resources" J. Netw. Syst. Manag. 26(2), 361–400, 2018

[5] R. Buyya, J. Broberg, and A. Goscinski, Eds., Cloud Computing. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2011. doi: https://doi.org/10.1002/9780470940105.

[6] G. Ghatikar, V. Ganti, N. Matson, and M. A. Piette, “Demand Response Opportunities and Enabling Technologies for Data Centers: Findings From Field Studies,” www.osti.gov, Aug. 01, 2012. https://www.osti.gov/servlets/purl/1174175./Accessed April. 24, 2023.

[7] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, “Recalibrating global data center energy-use estimates,” Science, vol. 367, no. 6481, pp. 984–986, Feb. 2020, doi: https://doi.org/10.1126/science.aba3758.

[8] C. S. Yeo and R. Buyya, “Service Level Agreement based Allocation of Cluster Resources: Handling Penalty to Enhance Utility,” Proceedings, Sep. 2005, doi: https://doi.org/10.1109/clustr.2005.347075.

[9] V. C. Emeakaroha, M. A. S. Netto, R. N. Calheiros, I. Brandic, R. Buyya, and C. A. F. De Rose, “Towards autonomic detection of SLA violations in Cloud infrastructures,” Future Generation Computer Systems, vol. 28, no. 7, pp. 1017–1029, Jul. 2012, doi: https://doi.org/10.1016/j.future.2011.08.018.

[10] R. Jain and N. Sharma, “A quantum inspired hybrid SSA–GWO algorithm for SLA based task scheduling to improve QoS parameter in cloud computing,” Cluster Computing, Sep. 2022, doi: https://doi.org/10.1007/s10586-022-03740-x.

[11] Z. S. Ageed and S. R. M. Zeebaree, “Distributed Systems Meet Cloud Computing: A Review of Convergence and Integration,” International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 11s, pp. 469–490, Jan. 2024, Accessed: Feb. 27, 2024.

[12] P. Jain and Sanjay Kumar Sharma, “A systematic review of nature inspired load balancing algorithm in heterogeneous cloud computing environment,” Nov. 2017, doi: https://doi.org/10.1109/infocomtech.2017.8340645.

[13] Cristiano, Luiz Fernando Bittencourt, T. Augusto, L. M. Peixoto, and E. Roberto, “RAaaS: Resource Allocation as a Service in multiple cloud providers,” Journal of Network and Computer Applications, vol. 221, pp. 103790–103790, Jan. 2024, doi: https://doi.org/10.1016/j.jnca.2023.103790.

[14] A. Beloglazov and R. 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 and Experience, vol. 24, no. 13, pp. 1397–1420, Oct. 2011, doi: https://doi.org/10.1002/cpe.1867.

[15] P. Jain and Sanjay Kumar Sharma, "Virtual Machine Consolidation Techniques to Reduce Energy Consumption in Cloud Data Centers: A Survey," Lecture notes in networks and systems, pp. 873–886, Jan. 2023, doi: https://doi.org/10.1007/978-981-99-5166-6_58..

[16] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," ACM SIGARCH Computer Architecture News, vol. 35, no. 2, pp. 13–23, Jun. 2007, doi: https://doi.org/10.1145/1273440.1250665.

[17] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Future Generation Computer Systems, vol. 28, no. 5, pp. 755–768, May 2012, doi: https://doi.org/10.1016/j.future.2011.04.017 .

[18] P. Jain and Sanjay Kumar Sharma, "A Load Balancing Aware Task Scheduling using Hybrid Firefly Salp Swarm Algorithm in Cloud Computing," International journal of computer networks and applications, vol. 10, no. 6, pp. 914–914, Dec. 2023, doi: https://doi.org/10.22247/ijcna/2023/223686.

[19] A. Choudhary, M. C. Govil, G. Singh, L. K. Awasthi, E. S. Pilli, and D. Kapil, "A critical survey of live virtual machine migration techniques," Journal of Cloud Computing, vol. 6, no. 1, Nov. 2017, doi: https://doi.org/10.1186/s13677-017-0092-1.

[20] N. Kr. Biswas, S. Banerjee, U. Biswas, and U. Ghosh, "An approach towards development of new linear regression prediction model for reduced energy consumption and SLA violation in the domain of green cloud computing," Sustainable Energy Technologies and Assessments, vol. 45, p. 101087, Jun. 2021, doi: https://doi.org/10.1016/j.seta.2021.101087.

[21] D. Minarolli, A. Mazrekaj, and B. Freisleben, "Tackling uncertainty in long-term predictions for host overload and underload detection in cloud computing," Journal of Cloud Computing, vol. 6, no. 1, Feb. 2017, doi: https://doi.org/10.1186/s13677-017-0074-3.

[22] S.B Melhem et al., " "Markov Prediction Model for Host Load Detection and VM Placement in Live Migration | IEEE Journals & Magazine | IEEE Xplore," ieeexplore.ieee.org. https://ieeexplore.ieee.org/abstract/document/8226661/ Accessed April 12, 2023.

[23] N. Patel and H. Patel, "Energy efficient strategy for placement of virtual machines selected from underloaded servers in compute Cloud," Journal of King Saud University - Computer and Information Sciences, vol. 32, no. 6, pp. 700–708, Jul. 2020, doi: https://doi.org/10.1016/j.jksuci.2017.11.003.

[24] Ferdaus, Md Hasanul, M. Murshed, R. N. Calheiros, and R. Buyya, "Multi-objective, Decentralized Dynamic Virtual Machine Consolidation using ACO Metaheuristic in Computing Clouds," arXiv (Cornell University), Jun. 2017, doi: https://doi.org/10.48550/arxiv.1706.06646

[25] W. Ding, F. Luo, L. Han, C. Gu, H. Lu, and J. Fuentes, "Adaptive virtual machine consolidation framework based on the performance-to-power ratio in cloud data centers," Future Generation Computer Systems, vol. 111, pp. 254–270, Oct. 2020, doi: https://doi.org/10.1016/j.future.2020.05.004.

[26] A. Abdelsamea, A. A. El-Moursy, E. E. Hemayed, and H. Eldeeb, "Virtual machine consolidation enhancement using hybrid regression algorithms," Egyptian Informatics Journal, vol. 18, no. 3, pp. 161–170, Nov. 2017, doi: https://doi.org/10.1016/j.eij.2016.12.002.

[27] Y. Saadi and S. El Kafhali, "Energy-efficient strategy for virtual machine consolidation in cloud environment," Soft Computing, Mar. 2020, doi: https://doi.org/10.1007/s00500-020-04839-2.

[28] W. Ding, F. Luo, L. Han, C. Gu, H. Lu, and J. Fuentes, "Adaptive virtual machine consolidation framework based on the performance-to-power ratio in cloud data centers," Future Generation Computer

Systems, vol. 111, pp. 254–270, Oct. 2020, doi: https://doi.org/10.1016/j.future.2020.05.004.

[29] U. Arshad, M. Aleem, G. Srivastava, and J. C.-W. Lin, "Utilizing power consumption and SLA violations using dynamic VM consolidation in cloud data centers," Renewable and Sustainable Energy Reviews, vol. 167, p. 112782, Oct. 2022, doi: https://doi.org/10.1016/j.rser.2022.112782

[30] L. Pellegrina and Fabio Vandin, "SILVAN : Estimating Betweenness Centralities with Progressive Sampling and Non-uniform Rademacher Bounds," ACM Transactions on Knowledge Discovery From Data, vol. 18, no. 3, pp. 1–55, Dec. 2023, doi: https://doi.org/10.1145/3628601.

[31] G. Dhiman, Kresimir Mihic, and T. Rosing, "A system for online power prediction in virtualized environments using Gaussian mixture models," Jun. 2010, doi: https://doi.org/10.1145/1837274.1837478