International Journal for Innovative
Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER
SSRN

IJIEMR Transactions, online available on 5th Oct 2021. Link :http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=ISSUE-10

DOI: 10.48047/IJIEMR/V10/I10/24

Title AN SYSTEM SOFTWARE APPROACH FOR CONSUMING OF ENERGY BASED ON KUBERNETES CLUSTERS IN CLOUD DATA CENTRES

Volume 10, Issue 10, Pages: 135-144

Paper Authors

T.prasanna, Dr.kishan pal singh

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

AN SYSTEM SOFTWARE APPROACH FOR CONSUMING OF ENERGY BASED ON KUBERNETES CLUSTERS IN CLOUD DATA CENTRES

[1]T.prasanna, [2]Dr.kishan pal singh
[1]Research scholar IET and science, Mangalyatan University, Aligarh
[2]Principal, Mangalyatan University.

Abstract: One of the reasons for the fast adoption of cloud computing for hosting services is the pay-as-you-go price model and other attractive features. When it comes to carbon emissions, cloud data centres are one of the most rapidly growing sources because of how much electricity they use. Increasing resource utilisation and utilising renewable energy sources are two effective methods for reducing resource waste and improving energy efficiency. Project's aim is to reduce the carbon footprint of data centres by utilising less brown and greener energy. Microservices and renewable energy are used to handle interactive and batch workload resources in our self-adaptive approach. To ensure the quality of service of workloads, a brownout-based strategy is recommended for interactive workloads and a delayed approach is recommended for batch workloads. With real-world online services as the test bed, we've discovered that the approach works. Results indicate that by implementing our approach, we can reduce brown energy consumption by 21% while boosting renewable energy consumption by 10%.

Keywords: Brownout, Cloud Data Centres, QoS, Microservices, Renewable Energy Efficiency

I. INTRODUCTION

Web services, in particular, which are mostly hosted in cloud data centres, are increasingly relied upon by companies and society in today's world. A major benefit of cloud computing for companies is that it removes the need for them to develop their own IT infrastructures, allowing them to concentrate on their core strengths instead of IT and infrastructure problems. Elasticity, availability, and the pay-as-you-go pricing model have all contributed to the growth of cloud computing. It was necessary to build numerous data centres across world since cloud computing was expanding so quickly. Many of these data centres were owned by large companies and cloud service providers such as

Amazon, Microsoft and Google A significant amount of energy is used by the cloud data centres, resulting in high operational costs and substantial carbon emissions [3].

IT now consumes around 7% of global energy, and that number is projected to grow to 13% by 2030 [4]. The operation of data centres [5] is one of the fastest growing sources of CO2 emissions. 91

billion kWh of power was used in 2013 by US data centres (equivalent to the country's two-year electricity consumption).

II. RELATED WORK

DVFS and VM consolidation

A significant amount of research on the energy efficiency of data \scenters has been devoted to the optimization techniques \sto decrease the energy consumption of servers inside a data scenter utilising technologies such as dynamic voltage and \sfrequency scaling (DVFS) and VM consolidation [17][18]. Liu et al [19] developed a heuristic method for big data \stask scheduling based on thermal-aware and DVFS-enabled \stechniques to reduce the overall energy consumption of \sdata centres. Kim et al. [18] analysed real-time service as VM \srequests and suggested various DVFS methods to reduce \senergy usage for the DVFS-enabled cluster. Cheng \set al. [27] presented a heterogeneity-aware task assignment \sapproach to reduce the total energy usage in \sa heterogeneous Hadoop cluster without compromising job \sperformance. It was shown by Teng et al. [20] that using DVFS and VM consolidation combined for batch-oriented situations yielded many heuristic methods. Nguyen et al. [21] \sintroduced a virtual machine consolidation method with \smultiple use prediction to enhance the energy efficiency \sof cloud data centres.

Our approach varies from previous attempts in many perspectives: (1) none of these DVFS-based and VM consolidation \sapproaches can operate effectively if the entire system is overloaded; (2) none of them made the efforts to plan the \smixed kind of workloads; (3) none of them used the \srenewable energy to power their systems.

Brownout

[13] A study and taxonomy on brownout-based methods presented by Xu and colleagues described the use of brownout in cloud computing systems for a variety of optimization purposes. [14] Brownout was used by Tomas et al. [28] to deal with cloud load balancing problems. By encouraging customers to smooth out their use variations, Shahrad et al. [29] developed a viable pricing model for brownout systems with the goal of increasing the utilisation of the cloud infrastructure. When it comes to implementing brownouts and turning off microservices, there are trade-offs to be made. [30], for example, examines the energy-revenue trade-off. Our optimization goal, on the other hand, is to control the amount of energy used in cloud data centres.

Brownout-based methods to managing microservices and resources from an energy viewpoint were described by Xu et al. [22][31]. Researchers Hasan et al. [23] looked into the green energy vs. user experience conflict in interactive cloud applications, and they developed a controller that ensures response time stays within the Service Level Agreement (SLA) range even when green energy is used, thanks to a brownout-enabled design. While earlier studies have focused on managing energy in a piecemeal fashion, this one incorporates a comprehensive approach that takes into account various levels of resource management, cooling power, and diverse workloads, as well as using renewable energy to cut down on brown energy use.

Renewable Energy

Studies in the literature have explored the optimization of data centre on-site renewable energy usage. As a research platform, Goiri et al [16] constructed a solar-powered data centre with battery storage and a grid-tie, and presented it.

A new technique for dynamically scheduling interactive and batch workloads, dubbed GreenSwitch, is also described, as is the energy source to be used. Using GreenSwitch, the total cost of energy is reduced while workload and battery life restrictions are respected. When compared to other researchers' work, ours is more of a multi-layer scheduling strategy that takes into account the application as well as VMs and hosts. In addition, we create models of the programmes that would be affected by the brownout. Furthermore, our support vector machine-based renewable energy prediction model varies substantially from their model, which projected solar energy based on the most recent period.

In the data centre, Liu et al. [25] looked at how to shift workloads and match renewable energy supply with demand. In a data centre, they plan for non-critical IT demand and distribute IT resources based on the availability of renewable power supply and cooling system efficiency. It was a restricted convex optimization problem with the goal of reducing the data center's total costs to the absolute minimum. The optimization of the energy viewpoint differs from the optimization of total expenses. Also, we can improve power consumption by scheduling both interactive and batch workloads, whereas [25] simply optimises batch workload scheduling without optimising interactive workloads.

In our earlier work [26], we focused on microservices management as finite Markov Decision Processes to maximise onsite renewable energy usage (MDP). To achieve a balance between workload execution and brown energy consumption, the suggested approach dynamically turns off non-mandatory microservices of the application. It also recommends how much battery power should be used throughout each time slot based on a greenness property rating. We, on the other hand, propose combined management of both interactive and batch workloads as an alternative to this practise. Also, we cover the full stack of resource scheduling, such as microservices, virtual machines, and physical hosts. We also put our system through its paces on a real testbed, while [26] just uses simulations.

The current study adds to the increasing body of knowledge in the relevant field by presenting new findings. Table 1 compares the relevant work based on essential methods, energy model, workload kinds, and resource scheduling layers, with the results shown in the table.

The main distinction between our proposed study and past works must be highlighted, given the contributions of previous works. According to what we know, this is the first time interactive workloads using brownout and batch workloads using deferral method have been managed together. Only one of these characteristics has been used in previous work. VM scheduling and host scaling with a genuine testbed are also taken into account as part of our multi-layer resource scheduling approach, which allows us to create a fully integrated resource management.

III. SYSTEM MODEL

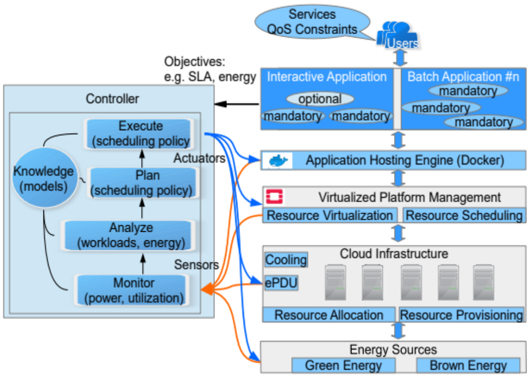


Fig. 1: Perspective Model

Figure 1 shows our system model for adaptive resource scheduling, which we describe in the next section. In the application layer, we examine both interactive and batch workloads, and in the energy supply layer, we consider green and brown energy together. Users send service requests to the system via the users' layers. When submitting a request, customers may provide QoS restrictions like a budget and a deadline. The application layer receives the given requests and processes them. These user-generated workloads are handled by apps housed in the cloud architecture from the perspective of service providers. Applications can be divided into two categories: interactive (like a web application) and batch.

To guarantee QoS, the interactive application should be run as soon as possible.. To us, the interactive application is a brownout supporter, and the microservices of interactive apps may be categorised as optional or essential. If required, the extra features may be disabled to economise on resource use. Tasks may be postponed to a certain day and time for batch applications. Application hosting engines like Kubernetes [34] or Apache Tomcat handle applications supplied by service providers to deliver services to consumers. Virtualized platforms (virtual resources) and cloud infrastructure are also options for deploying applications (physical resources). Container-based management systems, such as Kubernetes Swarm [35], Kubernetes [36], or Mesos [37], may be used as the host application engine. Numerous VMs may be used to launch an application with multiple microservices. It is the virtualized platform that oversees the virtualized resources such as VMware's Virtual Machines. Infrastructure management platforms such as OpenStack [39] may handle cloud infrastructure resource allocation and provisioning. It's possible to install a number of virtual machines on various hosts.

The energy supply layer, located at the bottom of the stack, is where the system gets its power from a variety of sources. This kind of brown energy is generated by a coal-fired thermal power station, which has a large carbon impact. Solar power, for example, is a renewable source of green energy. a controller based on the MAPE-K architecture model is required to support the provision of resources, monitoring, and allocation in the system, and it fits into the feedback loop of the MAPE-K process, which has modules such as Monitor, Analyze, Plan, and Execute to achieve the cloud computing system adaptation process [41]. Interactions with the system are established via the use of sensors and actuators. Information is gathered by sensors at many layers of the system, including the application hosting engine, virtualization platform, cloud infrastructure, and energy consumption, among others. Sensors may be hardware-attached devices like a power metre or a temperature sensor. The Monitor module receives the data that was gathered and uses it.

Modules Analyze and Plan create choices for applying scheduling rules in which scheduling policies are applied based on information obtained from Monitor and Analyze modules. Execute module schedules resources via actuators on application hosting engine and virtualized platform in accordance with the decisions to enable/disable optional microservices in interactive applications or defer the workloads of batch applications to be supplied by renewable energy, according to the decisions. A virtualized platform or an application hosting engine may do these tasks through Application Programming Interfaces (APIs).

Energy efficiency and SLA restrictions are only two examples of trade-offs that may be stored in the MAPE-K model's Knowledge pool. SLA rules, for example, may be modified based on resource allocation algorithms in the Knowledge Pool. Scheduling algorithms may make use of models like those in the Knowledge pool to estimate the quantity of renewable energy that will be delivered. Our methodology and prototype system will be discussed in detail in the parts that follow, starting with an overview of our viewpoint model.

IV. PROTOTYPE SYSTEM IMPLEMENTATION

In Section 3, we developed a system model and evaluated it by building a prototype system on our testbed. The prototype system's architecture is shown in Figure 3. As a result, we drew on established cloud resource management systems as well as microservices management platforms to design and build our prototype. OpenStack, a cloud IaaS resource management platform, is in charge of cloud resources such as CPU, memory, and bandwidth. OpenStack. Status collector collects resource monitoring data, which may then be utilised for resource allocation and optimisation. Using Kubernetes Swarm as a microservice management platform, it's possible to keep track of service images, monitor resource use, and manage service lifecycles all in one place. It's also possible to utilise other Kubernetes APIs to perform operations on services In the system architecture shown in Figure 1, these two platforms are translated to the Virtualized Platform Management and Application Hosting Engine layers.

Cloud resources and services are managed by SA (Self-Adaptive) controller, which is intended to manage and monitor both of them in order to accomplish multiple-level resource scheduling, as shown in Figure 1. Resource allocators in SA controller handle cloud resource management platform and service management platform concurrently to accept and process requests by delivering the required amount of resources when requests like interactive workloads or batch workloads are submitted to the system.

The resource allocator can do more than just allocate resources to requests; it can also optimise resource usage. If you want to minimise resource use, you may use brownout to disable unnecessary microservices. Service providers may also set up accessible energy-saving resource scheduling strategies. The resource monitor is required to gather resource use data at many levels, including services, VMs, and hosts, in order to provide and optimise resources using a resource allocator. The collection time intervals should be properly set by the service provider to reduce the overheads of regularly monitored data gathering. For example, since brownout costs are low, the brownout mechanism may be checked every five minutes. In contrast, VM migration and host scaling procedures, on the other hand, can be carried out at intervals as long as an hour.

Below that, we'll go through the specifics of how our prototype system works.

IMPLEMENTATION

We use the OpenStack cloud resource management platform and the KubernetesSwarm service management platform to build our prototype system. Java, OpenStack, Kubernetes Swarm, Ansible, and the Eaton Power Distribution Units (ePDU) API are used to build the system. These open source technologies are used in our prototype system to offer a self-adaptive approach to resource optimization, management, and provisioning for various workload types.

Hosts and VMs are managed by the OpenStack platform.

These machines are known together as compute nodes because they link a hypervisor and an OpenStack controller through the Nova Compute Node component. Create, migrate, and delete VM instances are all handled via the Nova API. OpenVSwitch and the Neutron OVS Agent work together to support the network.

The service offered by service providers is managed by Kubernetes Swarm Platform. In the service repository component, the pictures of services are kept, and the service repository component may get the images from distant to local sources. The service manager uses Kubernetes APIs to manage the services, which includes creating and deleting them. A service that keeps tabs on service usage and liveness keeps tabs on the status of services.

V. PERFORMANCE EVALUATION ENVIRONMENTAL SETTINGS

It has been shown in earlier research [31][17] that setting the upper and lower utilisation thresholds at 80 percent and 20 percent, respectively, allows us to achieve better energy consumption/quality of service trade-offs than other settings. Configuring the upper utilisation threshold below 80%, for example, may cause brownouts to occur too often, while setting it at 90% or above seldom causes brownouts. A 5-minute interval is also configured, as well as a 24-hour period for the whole schedule period We utilise Facebook's application execution traces gathered in October 2009 for batch workloads in the Hadoop environment7. The map phase of a task may take anywhere from 25-13000 seconds, depending on your configuration, while the reduction phase might take anywhere from 15-2600 seconds. A uniform distribution with = 6 hours and = 1 hour in N(, 2) generates the deadline for processing tasks. We also make the same assumption as in [16] about the workloads using the maximum cluster utilisation of 27%. As can be seen in Figure, the aforementioned workloads' normalised resource use over the course of one day. As a result, the workloads may be adjusted to fit within the availability of green energy. Even if green energy is available at hour 7, certain batch workloads may be postponed until a later time when there is more green energy available.

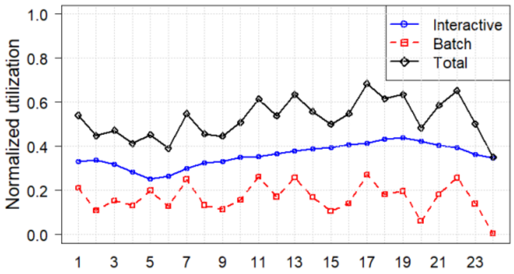


Fig. 4: Workloads distribution

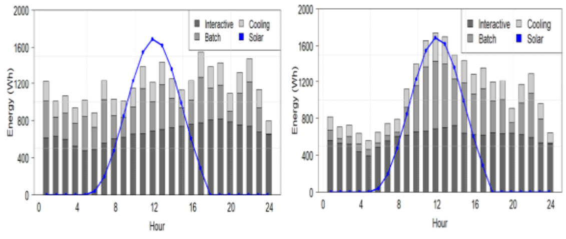
VI. RESULTS

VM consolidation based on Modified Best Fit Decreasing algorithm [17] that consolidates VMs on hosts that produce the least amount of energy incrementation and host scaling [45] that dynamically adds/removes hosts in the system based on profiling are used to evaluate the benefits of our proposed Green-aware and Self-adaptive approach (GSA) for renewable energy usage.Interactive workloads, batch workloads, and cooling energy usage are shown in Figure 5a during the observable time period (one day). The actual generation of renewable energy is shown by the blue line. For the city of Denver, the day is considered to be in the fall. There are approximately 12 hours of daylight during this time of year, which is less than during the summer but more than during winter. The system uses brown energy from 00:00 to 5:00 a.m. and 18:00 to 23:30 p.m. throughout the test day. During the day, from 6:00 to 17:00, solar energy is accessible.

When virtual machine consolidation and host scaling are used, the system's solar energy usage remains underutilised. It consumes about 1400 Wh at 11:00 a.m., whereas the available solar energy is more than 1500 W.

Figure shows the GSA approach's energy usage when Algorithms 1 to 4 are used. The blue line still depicts the actual generation of renewable electricity, but the choice is made using our SVM prediction model rather than historical data. By postponing operations until when solar energy is available, batch workloads use less power between the hours of 0:00 to 8:00, as can be shown by looking at the decreased power usage during this time period. Due to time limitations, certain batch workloads must still be run between the hours of 0:00 and 8:00 a.m. during non-renewable energy times. As a result, the brown energy consumption has decreased from 0:00 to 8:00 compared to Figure . When the clock strikes one o'clock, the total power drops to 815 Watts from 1221 Watts.

When solar energy is available, the GSA method has increased the use of renewable energy by following the projected energy consumption line. Solar energy use increases from 1387 to 1544 Wh during hour 11:00, as shown in Figure.



(a) (b)

Fig. 5: Results of (a) baseline HS (b) proposed approach GSA

With the average deactivation percentage, we can assess the effect of brownout on microservices, which is calculated by taking the total number of deactivated microservices during the observation period. As the average deactivation percentage rises, so do the number of microservices deactivated. Figures 8a and 8b show an average deactivation percentage of 11.2% and 10.8%, respectively, while Figure 8c shows a deactivation percentage of 10.1%. The findings show that batch workloads with a longer deadline and a longer daylight have fewer deactivated microservices. The rationale for this is that more active microservices may be supported by increased renewable energy availability.

VII. CONCLUSIONS AND FUTURE WORK

Self-adaptive management of applications and use of renewable energy in the cloud computing environment provide many possibilities to improve energy efficiency. For interactive and batch workloads, we developed a multiple-layer viewpoint model that takes renewable energy into account. As a result of the viewpoint model, we developed a self-adaptive and renewable energy-aware method. While maintaining the QoS requirements for workloads, the suggested method increases renewable energy use while reducing brown energy use. To estimate the amount of solar electricity generated in Denver City, we use a forecasting technique based on solar radiation. We then include this into our strategy. It is possible to dynamically disable/enable system components for interactive workloads using a brownout mechanism, while batch workloads employ a deferral method to delay execution until renewable energy is available. To decrease the number of active hosts, VM consolidation and host scaling are both used.

We created a working prototype so that we could test the efficacy of our strategy. OpenStack manages the prototype's physical resources, while Kubernetes Swarm handles the services. Monitoring, managing, and provisioning service resources are all made possible via the APIs provided by these platforms. The experimental evaluations using a microservices-based web system and workloads from actual traces demonstrate the efficacy of our suggested method. Research shows that our method may increase renewable energy use while meeting workload limitations.

We want to include a battery model into [16] in the future so that renewable energy can be stored and used more efficiently. To accommodate workload changes in data centres and reduce the worldwide carbon footprint, we also aim to expand our prototype system to include several clouds in various time zones.

By extending cloud services to the network's edge, fog and edge computing enhance the user's experience and system performance by decreasing latency. The IoT and edge devices, on the other hand, have power limitations, such as relying on batteries or harnessing renewable energy sources. As a result, energy should be used as effectively as possible. By temporarily disabling certain optional application components, the brownout technique may help these devices use less energy. We also plan to use the brownout method to mobile edge computing in the future to better manage the energy consumption of edge devices in the Internet of Things.

REFERENCES

[1] C. Kilcioglu, J. M. Rao, A. Kannan, and R. P. McAfee, "Usage patterns and the economics of the public cloud," in Proceedings of the 26th International Conference on World Wide Web, ser. WWW '17. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2017, pp. 83–91. [Online]. Available: https://doi.org/10.1145/3038912.3052707

[2] Y. Chen, X. Yang, Q. Lin, H. Zhang, F. Gao, Z. Xu, Y. Dang, D. Zhang, H. Dong, Y. Xu, H. Li, and Y. Kang, "Outage prediction and diagnosis for cloud service systems," in The World Wide Web Conference, ser. WWW '19. New York, NY, USA: ACM, 2019, pp. 2659–2665. [Online]. Available: http://doi.acm.org/10.1145/3308558.3313501

[3] W. Jiang, Z. Jia, S. Feng, F. Liu, and H. Jin, "Fine-grained warm water cooling for improving datacenter economy," in Proceedings of the 46th International Symposium on Computer Architecture, ser. ISCA '19. New York, NY, USA: ACM, 2019, pp. 474–486. [Online]. Available: http://doi.acm.org/10.1145/3307650.3322236

[4] M. Avgerinou, P. Bertoldi, and L. Castellazzi, "Trends in data centre energy consumption under the european code of conduct for data centre energy efficiency," Energies, vol. 10, no. 10, p. 1470, 2017.

[5] B. Whitehead, D. Andrews, A. Shah, and G. Maidment, "Assessing the environmental impact of data centres part 1: Background, energy use and metrics," Building and Environment, vol. 82, pp. 151 – 159, 2014.

[6] P. Delforge. (2014) Data center efficiency assessment - scaling up energy efficiency across the data center industry: Evaluating key drivers and barriers. [Online]. Available: https://www.infrastructureusa.org/ scaling-up-energy-efficiency-across-the-data-center-industry/ [7] M. Pedram, "Energy-efficient datacenters," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 31, no. 10, pp. 1465–1484, 2012.

[8] A. Beloglazov and R. Buyya, "Openstack neat: a framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds," Concurrency and Computation: Practice and Experience, vol. 27, no. 5, pp. 1310–1333, 2015.

[9] A. Beloglazov and R. Buyya, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service

constraints," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 7, pp. 1366–1379, 2013.

[10] B. Clement. (2017) Rightscale 2017 state of the cloud report uncovers cloud adoption trends. [Online]. Available: https://www.rightscale.com/press-releases/

[11] S. Newman, Building Microservices. " O'Reilly Media, Inc.", 2015. Authorized licensed use limited to: Heriot-Watt University. Downloaded on September 20,2020 [12] Z. Ren, W. Wang, G. Wu, C. Gao, W. Chen, J. Wei, and T. Huang, "Migrating web applications from monolithic structure to microservices architecture," in Proceedings of the Tenth AsiaPacific Symposium on Internetware, ser. Internetware '18. New York, NY, USA: ACM, 2018, pp. 7:1–7:10. [Online]. Available: http://doi.acm.org/10.1145/3275219.3275230

[13] M. Xu and R. Buyya, "Brownout approach for adaptive management of resources and applications in cloud computing systems: A taxonomy and future directions," ACM Computing Surveys, vol. 52, no. 1, pp. 8:1–8:27, 2019.

[14] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," in ACM SIGMETRICS Performance Evaluation Review, vol. 40, no. 1. ACM, 2012, pp. 175– 186.

[15] Z. Han, H. Tan, G. Chen, R. Wang, Y. Chen, and F. C. M. Lau, "Dynamic virtual machine management via approximate markov decision process," in Proceedings of the 35th Annual IEEE International Conference on Computer Communications (INFOCOM), April 2016, pp. 1–9.

[16] ´I. Goiri, W. Katsak, K. Le, T. D. Nguyen, and R. Bianchini, "Parasol and greenswitch: Managing datacenters powered by renewable energy," in ACM SIGARCH Computer Architecture News, vol. 41, no. 1. ACM, 2013, pp. 51–64.

[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, 2012.

[18] K. H. Kim, A. Beloglazov, and R. Buyya, "Power-aware provisioning of virtual machines for real-time cloud services," Concurrency and Computation: Practice and Experience, vol. 23, no. 13, pp. 1491– 1505, 2011.

[19] H. Liu, B. Liu, L. T. Yang, M. Lin, Y. Deng, K. Bilal, and S. U. Khan, "Thermal-aware and dvfs-enabled big data task scheduling for data centers," IEEE Transactions on Big Data, vol. 4, no. 2, pp. 177–190, 2018.

[20] F. Teng, L. Yu, T. Li, D. Deng, and F. Magoules, "Energy efficiency ` of vm consolidation in iaas clouds," The Journal of Supercomputing, pp. 1–28, 2016.

[21] Karthik, A., MazherIqbal, J.L. Efficient Speech Enhancement Using Recurrent Convolution Encoder and Decoder. Wireless Pers Commun (2021). https://doi.org/10.1007/s11277-021-08313-6