

Load Balanced and Energy Aware Cloud Resource Scheduling Design for Executing Data-intensive Application in SDVC

Ms. Shalini. S¹, Dr. Annapurna P Patil²

Department of Computer Science and Engineering, ACM Engineering College, Bangalore, India¹

Department of Computer Science and Engineering, Ramaiah Institute of Technology, Bangalore, India²

Abstract—Cloud computational platform provisions numerous cloud-based Vehicular Adhoc Network (VANET) applications. For providing better bandwidth and connectivity in dynamic manner, Software Defined VANET (SDVN) is developed. Using SDRVN, new VANET framework are modeled; for example, Software Defined Vehicular Cloud (SDVC). In SDVC, the vehicle enables virtualization technology through SDRVN and provides complex data-intensive workload execution in scalable and efficient manner. Vehicular Edge Computing (VEC) addresses various challenges of fifth generation (5G) workload applications performance and deadline requirement. VEC aid in reducing response time, delay with high reliability for workload execution. Here the workload tasks are executed to nearby edge devices connected to Road Side Unit (RSU) with limited computing capability. If the resources are not available in RSU, then the task execution is offloaded through SDN toward heterogeneous cloud server. Existing workload scheduling in cloud environment are designed considering minimizing cost and delay; however, very limited work has been done considering energy minimization for workload execution. This paper presents a Load Balanced and Energy Aware Cloud Resource Scheduling (LBEACRS) design for heterogeneous cloud framework. Experiment outcome shows the LBEACRS achieves better makespan and energy efficiency performance when compared with standard cloud resource scheduling design.

Keywords—Cloud computing; data-intensive applications; heterogenous server; IEEE 802.11p; software defined network; software defined vehicular cloud; vehicular adhoc network; workload scheduling; road side unit; vehicular edge cloud

I. INTRODUCTION

As of late, its seen that the amount of Internet associated gadgets are more when compared with the quantity of people in the world, Internet-associated gadgets are expected to exceed thirty billion by 2020, as per report suggested in [1], thus, accidentally apprehending the IoT models. As there is an enormous increase in the vehicles which are connected to the internet, the normal VANETs are merged into these Internet of Vehicles (IoV). In the Internet of Vehicles period, smart devices and sensors are connected to the vehicles which can give smart vehicle maneuvering, video streaming, prevention of accidents, traffic management, navigating capabilities, just as a large group of arising intelligent applications, a particularly augmented and virtual reality. A vast majority of these applications require complicated computation and various methods to recognize the patterns, which can compute

intensive data and thus need a processor which can provide a powerful and dedicated virtual machine for the computation. The restricted capability for the computation and the because of less resource's capacity in the vehicles, it is a challenging task for the decision-making, networking and data processing in real-time. Due to this problem, it develops an issue for computing the data and allocating the resources to the different applications in the VANETs which have limited resources.

To resolve the problems of unstable computation in the vehicles, the usage of the vehicle nodes is suggested. For the improvement of the safety and comfort for the travelers in a cloud-based vehicle network a method using the Road Side Units (RSUs) can be used. In the RSU, using the computing methods and an integrated communication technology, various offload task can be done using the cloud, hence this reduces the power consumption and the storage capacity in the Onboard Unit which is present in the vehicles. This method is called as Mobile Cloud-Computing (MCC), which provides an improved utilization of the resource, better performance for computation and also gives many benefits, yet not restricted to, 1) increasing the lifetime performance of the battery by offloading the consumption of energy to the cloud, 2) empowering complex memory exploiting gadgets to portable clients, and 3) giving more memory-storage to the clients. Nonetheless, thinking about the limit constraint and fluctuation in the delay for transmitting the data to backbone and backhaul systems [2], placing the cloud network far from the vehicular network can reduce the efficiency of the offload. All the things considered, the vehicle edge computing (VEC) has been given which places the cloud networks near the edge of the radio access organization, to be specific, near the Road Side Units, which gives the computation results within the range of communication with the help of SDN.

The concentrated idea of Vehicular Edge Computing presents critical difficulties particularly in a profoundly unique condition like VANET. Besides, in vehicles having high mobility [3], [4], particularly in roads encountering more traffic, requires an appropriate scheduling to prepare the offloaded complicated workload assignments which can oblige the tasks which are delay tolerant (for example safety related application, accident prevention) just as computationally thorough assignments (for example video observation and dynamic image/video-based applications). The intercommunication requests for quality of experience (QoE) and quality of services (QoS) to understand the related

advantages like maximized throughput, low latency, fast communication, high reliability and hence forth. These prerequisites makes workload resource scheduling more testing in vehicular cloud environment [5].

In addressing above discussed research problem and challenges existing workload scheduling methods either focused on minimizing energy or makespan individually; and very limited work have been emphasized in building scheduling optimization considering both together. However, existing model failed balance load for scheduling of newly arrived task. As a result, induce scheduling delay and increases operation cost because of higher SLA violation. In addressing performance issues the future design must incorporate effective load balancing technique with energy-makespan tradeoff requirement as a major objective. This motivates the research work to develop an improved scheduling designing incorporating effective load-balancing technique namely load balanced energy aware cloud resource scheduling. The LBEACRS design present an efficient design to schedule backlog task with high resource utilization. Further, for selecting cloud resource to execute workload a tradeoffs metric of energy maximization with high resource utilization meeting task deadline is presented.

The manuscript significance is described below:

- This paper presents a load balanced and energy aware cloud resource scheduling method for executing data-intensive workload on software defined vehicular cloud (SDVC).
- The SDN is used to offload workload execution to cloud environment. It also maintains routing table for delivering the workload execution outcome to respective vehicle users.
- The LBEACRS model is designed in such a way that it can balance load among different server, maximize resource utilization and reduce energy consumption by running less number of server and also guarantee dead requirement of workload with high makespan efficiency.
- The LBEACRS reduce energy consumption and achieve better makespan performance in comparison with existing DCOH workload scheduling methodology.

The manuscript is articulated as follow. In Section II, survey of existing model and limitation is described. In Section III the present propose methodology for workload scheduling in SDN enabled VANET-Cloud environment. In Section IV, experiment analysis is presented. Finally, the research work of LBEACRS is concluded with future research direction.

II. LITERATURE SURVEY

This section carry out critical review of various workload scheduling technique in reducing cost, makespan with deadline constraint; however, very limited work has been done to minimize energy constraint for workload execution in heterogeneous cloud environment [6]. In [6] presented energy

aware scheduling strategy for executing workload and showed heterogeneous computing platform provide multiple processing core for executing large-scale workload. However, these strategies induce high computation overhead and induce energy overhead considering different levels of computational process and storage operations [7]. In [8], analysis on a super personal computer has been performed which has 16 thousand nodes utilizes more energy. The consumption of energy is a critical problem which has a huge impact in the computation of the data and for the improvement of the system. In a heterogeneous cloud computing (HCC) condition, Directed Acyclic Graph (DAG) is used by an application which is running parallelly to allocate the jobs according to the priority. Moreover, the Directed Acyclic graph represents the edges and jobs which portrays the messages correspondingly to the different jobs [9]. In [10], it showed the critical difficulties exist for responsibility planning in a hybrid cloud environment. These are distinctive Cloud Service Providers (CSP), heterogeneous workload, which tell how to send and port the administration in the cloud environment having negligible financial spending plan. For guaranteeing prevalent asset usage they introduced a heterogeneous workload planning for the remote cloud environment. Additionally, to guarantee the execution of the workload, the execution is finished inside less time limitation a workload planning component is introduced utilizing the Backhaul Propagation Neural-Network in the hybrid cloud framework. In [11], recommended that the work process scheduling configuration considers the financial spending plan and time required for the computation in a hybrid cloud environment. The first scheduling-algorithm is planned utilizing the single-objective which works specifically for the DCOH technique. This technique is planned to decrease the monetary spending plan of workload scheduling with cut-off time essential. After this, they have introduced multi-objective on the basis of workload scheduled strategy to the specific MOH technique. This technique is intended to bring the trade-offs among the financial spending plan and execution time for workload execution. Notwithstanding, this technique is not much productive to reduce the energy for the logical work process computation in heterogenous cloud network. In Cloud network, which provides service to the market, the clients aid the cloud administrations given by CSP to perform the execution of the workload. Workload for the most part includes certain cut-off time essential for guarantying QoS. Simultaneously, the poor performance of QoS can force exacting a penalty on the CSP. Notwithstanding, the existing algorithm for workload scheduling accepts the time for the makespan of the tasks in the data concentrated workload application are fixed. In any case, this theory is for the cloud servers which have efficiently begun to help in the optimization of the energy, which is not utilized in the workload-scheduling model [11], [12].

From extensive study it can be seen existing cloud resource management technique either focused on reducing energy or makespan. Very few model presented to optimize time and cost together employing multi-objective optimization. Further, these model fails to meet dynamic load requirement of workload application; thus, limit the adoption dynamic workload application real-time requirement; in addressing an effective

load balancing mechanism is needed which is presented in next section.

III. LOAD BALANCED AND ENERGY AWARE CLOUD RESOURCE SCHEDULING DESIGN FOR TASK EXECUTION IN SDN ENABLED VANET-CLOUD PLATFORM

This section presents load balanced energy aware cloud resource scheduling design for SDN enabled VANET-Cloud. First, we describe the system model for SDN enabled VANET-Cloud. Second, presents load balanced and energy aware cloud resource Scheduling workload execution in SDN enabled VANET-Cloud. The software architecture of the defined VANET-Cloud is shown in Fig. 1.

A. System Model of SDN enabled VANET-Cloud

This section present system model used for task scheduling in SDN enabled VANET-Cloud. Here the vehicle moves at varying speed in particular direction and execute different kind of data intensive and scientific applications. Here the vehicles are connected to RSU which are placed in fixed position. These RSU are connected to SDN controllers through which task execution process is outsourced to cloud computing environment. The work aimed at minimizing task execution time with minimal energy consumption aiding better resource utilization through better load balancing strategy. The cloud environment used in this work is heterogeneous in nature which has different energy consumption and processing capability for executing different task.

B. Load Balanced and Energy Aware Cloud Resource Scheduling Design for Workload Execution in SDN enabled VANET-Cloud

This section present load balanced energy aware cloud resource scheduling (LBEACRS) method for workload execution in SDN enabled VNET-Cloud environment. Here the workload task submitted by vehicle is submitted to near-by RSU. The RSU execute the workload if it has resource available or it is offloaded to heterogeneous cloud server connected through SDN controller. The LBEACRS method is designed in a way that can optimally distribute the task with minimal consumption of energy and meeting task deadline constraint without overloading the particular server. As this work considers dispatching task to multi-server environment here a task backlog scheduling model is presented. A backlog scheduling model for a set of o multi-server platform T_1, T_2, \dots, T_o with size n_1, n_2, \dots, n_o and its processing speed is t_1, t_2, \dots, t_o is modelled in this work. Let consider that heterogeneous multi-server platform T_j is composed of n_j similar servers with processing capability t_j .

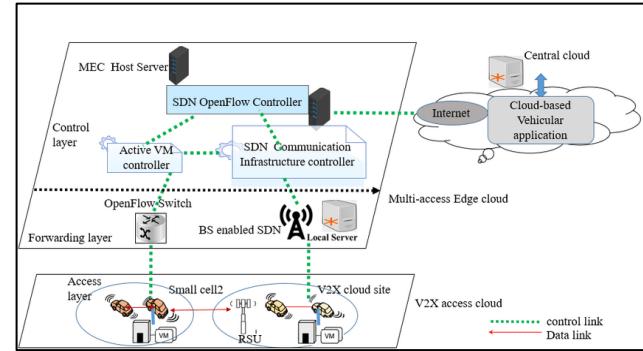


Fig. 1. Architecture of Software Defined VANET-Cloud.

Let consider a Poisson process with $M/M/m$ backlog scheduling model considering sequence of tasks with incoming load α . The incoming load is identical and independent in nature which distributed exponentially randomly (s) with average (\bar{s}) $1/\alpha$. The LBEACRS method divide these task sequence into o sub-sequences, in such way that the j^{th} subsequences with incoming load α_j is transmitted to multi-server platform T_j , where $1 \leq j \leq o$, $\alpha = \alpha_1 + \alpha_2 + \dots + \alpha_o$. A multi-server platform T_j keeps a buffer with unbounded capability for waiting task when entire server n_j is busy. Here the tasks are executed in first come first basis considering multi-server platform with task execution are identical and independent with exponential random parameter s and average \bar{s} . The n_j servers of multi-server platform T_j possess identical execution processing capability t_j ; thus, task processing makespan on respective sever of multi-server platform is identical and independent with exponential randomness as follows.

$$y_j = \frac{s}{t_j} \quad (1)$$

with average

$$\bar{y}_j = \frac{\bar{s}}{t_j} \quad (2)$$

The average amount of task that can be completed by respective server within T_j (i.e., average workload execution rate) is obtained using following equation.

$$\beta_j = \frac{1}{\bar{y}_j} \quad (3)$$

The server utilization (i.e., the mean percentage of time a server will be busy) is obtained using following equation.

$$\gamma_j = \frac{\alpha_j}{n_j \beta_j} = \frac{\alpha_j \bar{y}_j}{n_j} = \frac{\alpha_j \bar{s}}{n_j t_j}, \quad (4)$$

Let $p_{j,l}$ represent the probability that l task will be processed or in waiting in backlog system considering multi-server platform T_j and is obtained using following equation.

$$p_{j,l} = \begin{cases} p_{j,0} \frac{(n_j \gamma_j)^l}{l!}, & l < n_j; \\ p_{j,0} \frac{n_j^{n_j} \gamma_j^l}{l!}, & l \geq n_j; \end{cases} \quad (5)$$

where

$$p_{j,0} = \left(\sum_{l=0}^{n_j-1} \frac{(n_j \gamma_j)^l}{l!} + \frac{(n_j \gamma_j)^{n_j}}{n_j!} \cdot \frac{1}{1-\gamma_j} \right)^{-1} \quad (6)$$

The probability of freshly arrived workload task that must be waited i.e. (backlogged) in multi-server platform T_j when entire server in T_j is busy is computed using following equation.

$$P_{r,j} = \frac{q_j n_j}{1-\gamma_j} = p_{j,0} \frac{n_j^{n_j}}{n_j!} \cdot \frac{\gamma_j^{n_j}}{1-\gamma_j} \quad (7)$$

The mean amount of workload task that are being in execution processes or waiting within multi-server platform T_j is computed using following equation.

$$\bar{O}_j = \sum_{l=0}^{\infty} l p_{j,l} = n_j \gamma_j + \frac{\gamma_j}{1-\gamma_j} P_{r,j}. \quad (8)$$

Similarly, the mean workload task response makespan of multi-server platform T_j is computed using following equation.

$$U_j = \frac{\bar{O}_j}{\alpha_j} = \bar{y}_j + \frac{P_{r,j}}{n_j(1-\gamma_j)} \bar{y}_j = \bar{y}_j \left(1 + \frac{P_{r,j}}{n_j(1-\gamma_j)} \right) \quad (9)$$

For simplicity the mean workload task response makespan of multi-server platform T_j is computed using following equation.

$$U_j = \frac{\bar{s}}{t_j} \left(1 + p_{j,0} \frac{n_j^{n_j-1}}{n_j!} \cdot \frac{\gamma_j^{n_j}}{(1-\gamma_j)^2} \right) \quad (10)$$

The energy consumed for executing task is obtained using following equation.

$$Q = \alpha \mathcal{C} V^2 \mathcal{F} = \delta t^\mu$$

where α defines the task activity features, V defines voltage, \mathcal{C} represent load capacitane, and \mathcal{F} defines clock frequency, t defines processor execution speed. The parameter δ is computed using following equation.

$$\delta = \frac{\alpha \beta^2 \mathcal{C}}{\mathcal{C}^{2\rho+1}} \quad (11)$$

where β and ρ are some constant greater than 0. The parameter μ is computed using following equation.

$$\mu = 2\rho + 1. \quad (12)$$

This work consider heterogeneous multi-server platform; thus, the value of δ and μ is different for different server. Here we consider two different energy mode such as idle and active mode. In idle mode, the machine dosnt execute any workload task and energy consumed is obtained using following equation.

$$Q_j = n_j (\gamma_j \delta_j t_j^{\mu_j} + Q_j^*) = \alpha_j \bar{t} \delta_j t_j^{\mu_j-1} + n_j Q_j^*. \quad (13)$$

Similarly, in active mode the server is running and waiting for incoming workload task and energy consumed is obtained using following equation.

$$Q_j = n_j (\delta_j t_j^{\mu_j} + Q_j^*) \quad (14)$$

This work aimed at allocating ideal resource with minimal execution time for executing workload task under multi-server platform with varying processing speed and power consumption. Let consider a o number of multi-server cloud platform with size of n_1, n_2, \dots, n_o , with varying power consumption and processing capability for executing workload with requirement \bar{s} with task arrival rate α , and establish load distribution $\alpha_1, \alpha_2, \dots, \alpha_o$ in achieving high performance efficiency is obtained by minimizing following equation.

$$\min U(\alpha_1, \alpha_2, \dots, \alpha_o) \quad (15)$$

The above equation is subject to following constraint

$$G(\alpha_1, \alpha_2, \dots, \alpha_o) = \alpha, \quad (16)$$

where

$$G(\alpha_1, \alpha_2, \dots, \alpha_o) = \alpha_1 + \alpha_2 + \dots + \alpha_o, \quad (17)$$

and $\gamma_j < 1, \forall 1 \leq j \leq o$. Here the workload task are scheduled by minimizing Eq. (15) and meeting constraint defined in Eq. (16) and (17) in order to achieves high resource utilization and performance efficiency with minimal energy dissipation.

IV. SIMULATION ANALYSIS AND RESULTS

Here experiment is conducted for evaluating LBEACRS algorithm methodology over standard cloud resource scheduling algorithm [11], [12]. Energy efficiency and makespan are performance metrics used for validating performance. For carrying out experiment SIMITS simulator [13] is used. Further, for incorporating software defined vehicular cloud architecture CloudSimSDN [14] which is an extension of CloudSim simulator which is incorporated into SIMITS. Further for providing secure communication among communicating device such as vehicle, RSU, and SDN controller the message in SDN enabled vehicular adhoc network are encrypted using elliptical curve cryptography. Further, this assumes that each device is composed of tamper proof device obtained from trust authority for carrying out authentication and pseudo-identity-based data signing that assures anonymity of data owner and secure communication. For evaluating performance experiment is conducted by considering dynamic radio propagation model where vehicle will move from urban-to-rural-to-highway and vice versa. Experiment is conducted considering two workload such as Montage and CyberShake workload [15]. The Montage and CyberShake workflow have been used for experiment. The Montage workflow requires high I/O resource; however, the CyberShake requires CPU and memory resource. The simulation parameter considers for carrying out experiment are described Table I.

Fig. 2 shows the graphical representation of how the montage workflow is stored. Similarly in Fig. 3 the graphical representation of the CyberShake workflow has been represented.

A. Makespan Performance Evaluation

Here makespan performance is evaluated for both proposed LBEACRS and existing DCOH workload scheduling methodology. Two different workloads such as Montage and

CyberShake is considered for evaluation. First experiment is conducted using Montage workload where the job size is varied and makespan induced for executing workload using DCOH and LBEACRS is graphically shown in Fig. 4. The LBEACRS improves makespan performance by 66.955%, 81.48%, 86.14%, and 89.83% over DCOH when job size is 25, 50, 100, and 1000, respectively. An average makespan performance enhancement of 81.1% is achieved using LBEACRS over DCOH for executing Montage workload.

TABLE I. SIMULATION PARAMETERS CONSIDERED

Network Parameter	Value
SDN enabled vehicular Adhoc network Size	50km * 50km
Number of Vehicles	40
Number of RSU	1 per region
Modulation scheme	QAM-64
Mobility of devices	3 cycle per frame
SDN enabled vehicular Adhoc network coding rate	0.75
SDN enabled vehicular Adhoc network bandwidth	27 Mbps
SDN enabled vehicular Adhoc network data channel size	6
SDN enabled vehicular Adhoc network control channel size	1
Number of SDN switches per RSU	1
SDN enabled vehicular Adhoc network time slot size	$8 \mu s$
Message information size	27 bytes
Radio propagation mobility model	Dynamic environment such as urban, rural, and highway
MAC used	ENCCMA, TECA & ERS
Resource scheduling used	LBEACRS & DCOH
Workload used	Montage & CyberShake

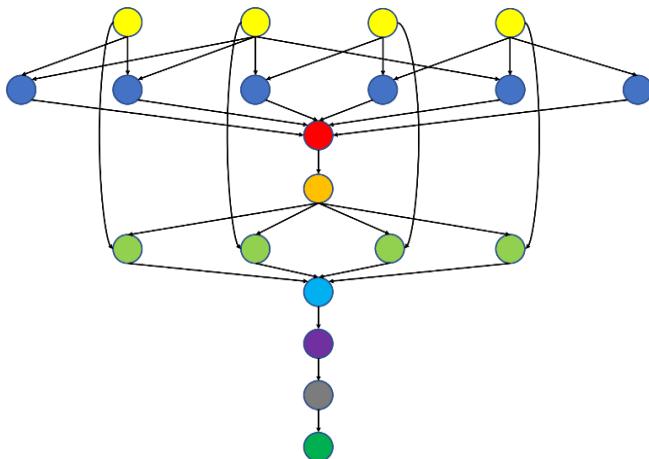


Fig. 2. A Sample Graphical Representation of Montage Workflow.

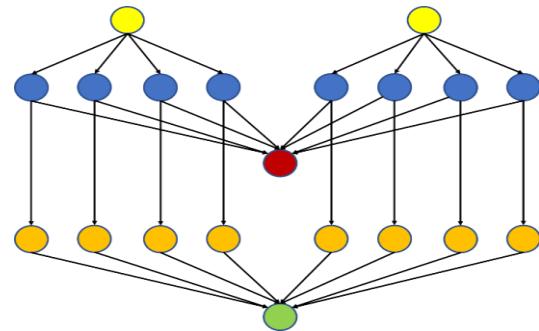


Fig. 3. A Sample Graphical Representation of CyberShake Workflow.

Montage workload execution makespan performance

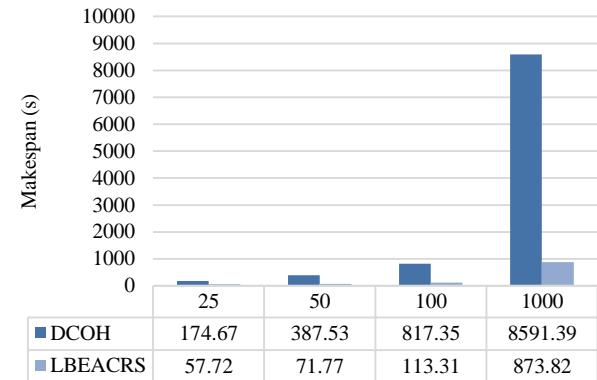


Fig. 4. Makespan Performance for Executing Montage Workload with different Workload Size.

Similarly, experiment is conducted using CyberShake workload where the job size is varied and makespan induced for executing using DCOH and LBEACRS is graphically shown in Fig. 5. The LBEACRS improves makespan performance by 69.95%, 82.14%, 89.23%, and 92.29% over DCOH when job size is 30, 50, 100, and 1000, respectively. An average makespan performance enhancement of 83.4% is achieved using LBEACRS over DCOH for executing CyberShake workload.

CyberShake workload excution makespan performance

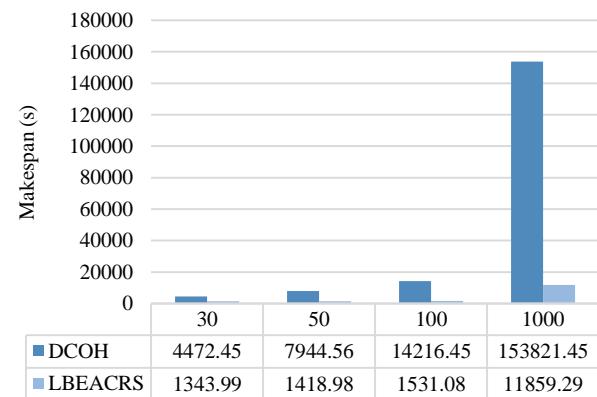


Fig. 5. Makespan Performance for Executing CyberShake Workload with different Workload Size.

B. Energy Consumption Performance Evaluation

Here energy performance is evaluated for both proposed LBEACRS and existing DCOH workload scheduling methodology. Two different workloads such as Montage and CyberShake is considered for evaluation. First experiment is conducted using Montage workload where the job size is varied and total energy induced for executing workload using DCOH and LBEACRS is graphically shown in Fig. 6. The LBEACRS improves energy efficiency performance by 74.63, 85.78%, 89.36%, and 92.19% over DCOH when job size is 25, 50, 100, and 1000, respectively. An average total energy consumption reduction of 85.49% is achieved using LBEACRS over DCOH for executing Montage workload. Further, the average energy incurred for executing each sub-task using DCOH and LBEACRS is graphically shown in Fig. 8. The LBEACRS improves average energy induced per task performance by 23.229% over DCOH for executing Montage workload.

Similarly, experiment is conducted using CyberShake workload where the job size is varied and total energy induced for executing workload using DCOH and LBEACRS is graphically shown in Fig. 7. The LBEACRS improves energy efficiency performance by 75.17%, 95.30%, 96.12%, and 93.63% over DCOH when job size is 30, 50, 100, and 1000, respectively. An average total energy consumption reduction of 90.057% is achieved using LBEACRS over DCOH for executing CyberShake workload. Further, the average energy incurred for executing each sub-task using DCOH and LBEACRS is graphically shown in Fig. 9. The LBEACRS improves average energy induced per task performance by 41.35% over DCOH for executing CyberShake workload.

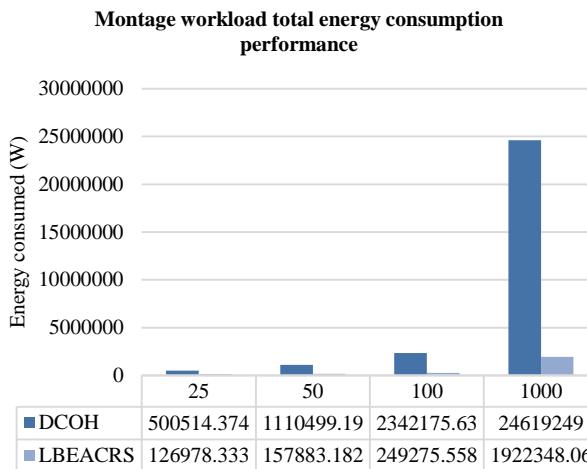


Fig. 6. Energy Consumption for Montage Workload Execution with different Workload Size.

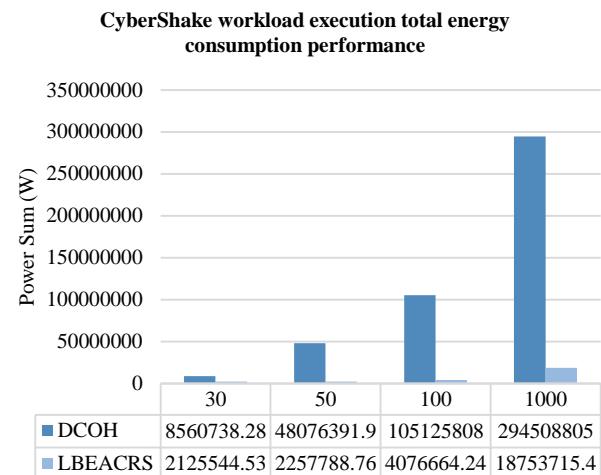


Fig. 7. Energy Consumption for CyberShake Workload Execution with different Workload Size.

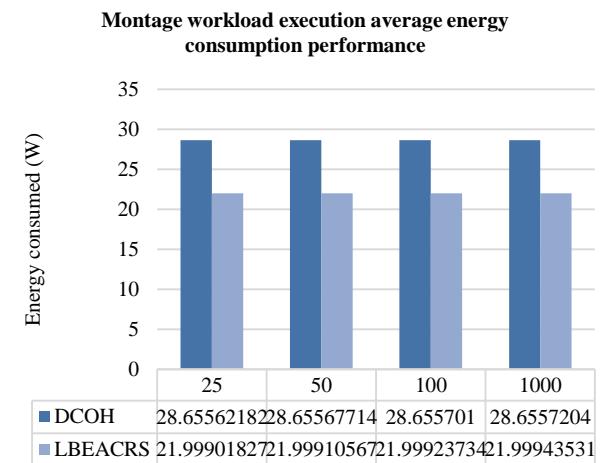


Fig. 8. Average Energy Consumption for Montage Workload Execution with different Workload Size.

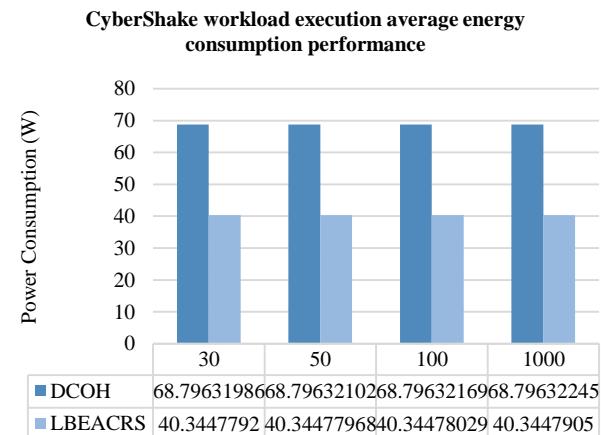


Fig. 9. Average Energy Consumption for CyberShake Workload Execution with different Workload Size.

From overall result attained it can be stated LBEACRS is scalable with respect to varying workload size and works well for both small and larger workload. And also shows they are efficient in provisioning CPU, memory and I/O intensive task.

V. CONCLUSION

This work studies the challenges involved in workload scheduling of data-intensive application on SDN enabled VANET-Cloud environment. From study we noted most of existing workload scheduling are done by predominantly considering minimizing makespan and also minimizing cost with deadline constraint. Further, doesn't balance load among processing node. In standard the cost in generally computed based on time spent. However, reducing energy plays major role as different countries has different prices. Thus, here a load balanced energy aware cloud resource scheduling is presented. Here the tasks are scheduled to node that consumes less energy with better resource utilization. Experiments are conducted using two workloads, namely Montage and CyberShake. These workloads are CPU, I/O, and memory intensive in nature. An average makespan performance enhancement of 81.1% and 83.4% is achieved using LBEACRS over DCOH for executing Montage and CyberShake workload, respectively. An average total energy consumption reduction of 85.49% and 90.057% is achieved using LBEACRS over DCOH for executing Montage and CyberShake workload, respectively. The LBEACRS improves average energy induced per task performance by 23.229% and 41.35% over DCOH for executing Montage and CyberShake workload, respectively. From overall result we can state that the LBEACRS is scalable irrespective of workload complexity and is robust with respect to CPU, I/O and memory intensive application.

Future work would consider introducing SLA with quality of experience guarantee in proving resource to VANET users. Further, incorporate effective security mechanism that can reduce key management, storage, and computation overhead.

REFERENCES

- [1] K. Chopra, K. Gupta and A. Lambora, "Future Internet: The Internet of Things-A Literature Review," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 135-139, doi: 10.1109/COMITCon.2019.8862269.
- [2] N. Abbas, Y. Zhang, A. Taherkordi and T. Skeie, "Mobile Edge Computing: A Survey," in IEEE Internet of Things Journal, vol. 5, no. 1, pp. 450-465, Feb. 2018, doi: 10.1109/JIOT.2017.2750180.
- [3] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues, and future challenges," IEEE Commun. Surveys Tuts., vol. 21, no. 2, pp. 12751313, 2nd Quart., 2019.
- [4] R. Lu, L. Zhang, J. Ni and Y. Fang, "5G Vehicle-to-Everything Services: Gearing Up for Security and Privacy," in Proceedings of the IEEE, vol. 108, no. 2, pp. 373-389, Feb. 2020, doi: 10.1109/JPROC.2019.2948302.
- [5] I. Din, B.-S. Kim, S. Hassan, M. Guizani, M. Atiquzzaman, and J. Rodrigues, "Information-centric network-based vehicular communications: Overview and research opportunities," Sensors, vol. 18, no. 11, p. 3957, Nov. 2018.
- [6] G. Xie, G. Zeng, R. Li and K. Li, "Energy-Aware Processor Merging Algorithms for Deadline Constrained Parallel Applications in Heterogeneous Cloud Computing," in IEEE Transactions on Sustainable Computing, vol. 2, no. 2, pp. 62-75, 1 April-June 2017.
- [7] Z. Li, J. Ge, H. Hu, W. Song, H. Hu and B. Luo, "Cost and Energy Aware Scheduling Algorithm for Scientific Workflows with Deadline Constraint in Clouds," in IEEE Transactions on Services Computing, vol. 11, no. 4, pp. 713-726, 1 July-Aug. 2018.
- [8] K. Li, "Power and performance management for parallel computations in clouds and data centers," J. Comput. Syst. Sci., vol. 82, no. 2, pp. 174-190, Mar. 2016.
- [9] G. Xie, L. Liu, L. Yang, and R. Li, "Scheduling trade-off of dynamic multiple parallel workflows on heterogeneous distributed computing systems," Concurrency Comput.-Practice Exp., vol. 29, no. 8, pp. 1-18, Jan. 2017.
- [10] Chunlin, L., Jianhang, T. & Youlong, L., "Hybrid Cloud Adaptive Scheduling Strategy for Heterogeneous Workloads", J Grid Computing (2019) 17: 419. <https://doi.org/10.1007/s10723-019-09481-3>.
- [11] Junlong Zhou et al., Cost and makespan-aware workflow scheduling in hybrid clouds. <https://doi.org/10.1016/j.sysarc.2019.08.004>, 2019.
- [12] Neelima, P., Reddy, A.R.M. An efficient load balancing system using adaptive dragonfly algorithm in cloud computing. Cluster Comput 23, 2891–2899, 2020.
- [13] Mario Manzano, Felipe Espinosa; Ning Lu; Xuemin Shen; Mark, J.W.; Fuqiang Liu, "Cognitive Self-Scheduled Mechanism for Access Control in Noisy Vehicular Ad Hoc Networks." Hindawi Publishing Corporation, Mathematical Problems in Engineering, Volume 2015, Article ID 354292, 2015.
- [14] J. Son, A. V. Dastjerdi, R. N. Calheiros, X. Ji, Y. Yoon and R. Buyya, "CloudSimSDN: Modeling and Simulation of Software-Defined Cloud Data Centers," 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Shenzhen, 2015, pp. 475-484.
- [15] Z. Zhu, G. Zhang, M. Li and X. Liu, "Evolutionary Multi-Objective Workflow Scheduling in Cloud," in IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 5, pp. 1344-1357, 1 May 2016.