A Dynamic Predictive VM Resource Scaling Strategy in Satellite-Ground Computing Networks

Siyan Pan
University of Chinese Academy of Sciences and Key Laboratory of Space Utilization, Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing, China
pansiyan18@csu.ac.cn

Suzhi Cao
Key Laboratory of Space Utilization, Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing, China
caosuzhi@csu.ac.cn

Lei Yan
Key Laboratory of Space Utilization, Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing, China
yanlei@csu.ac.cn

Houpeng Wang
University of Chinese Academy of Sciences and Key Laboratory of Space Utilization, Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences, Beijing, China
wanghoupeng19@mails.ucas.ac.cn

ABSTRACT

Combining satellite-ground network with the edge computing, an emerging research direction is to use low-orbit satellites as edge nodes to provide computing services for ground users and space missions. Due to the motion of satellites around the earth, the ground region covered by the satellite changes constantly over time, and the service traffic also changes accordingly. Therefore, the method of running a constant computing resource will lead to insufficient service capacity or high energy consumption. In this paper, we proposed a two-step dynamic resource management strategy SRTMS, which makes use of the certainty of satellite orbit and historical service data to predict the business traffic of future service region and dynamically scale the amount of in-orbit virtual computing resources. Through the strategy, energy consumption is reduced by 73% compared to the traditional mode in which all resources are operated at full capacity, saving resources that can be used for other payloads.

CCS CONCEPTS

• **Human-centered computing**; • **Ubiquitous and mobile computing**; • **Ubiquitous and mobile computing theory, concepts and paradigms**; • **Mobile computing**;

KEYWORDS

Satellite-ground network, VM resource management, Edge computing, Dynamic scaling strategy

CSAE 2021, October 19–21, 2021, Sanya, China

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487145

ACM Reference Format:
Siyan Pan, Suzhi Cao, Lei Yan, and Houpeng Wang. 2021. A Dynamic Predictive VM Resource Scaling Strategy in Satellite-Ground Computing Networks. In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3487075.3487145

1 INTRODUCTION

In order to adapt to the rapid development of satellite networks and optimize the service quality of on-orbit satellites, the research on combining ground edge computing technology with spatial information networks has attracted much attention. Compared with the traditional satellite network [1], where the satellite is only used for data forwarding, the core of the research of satellite edge computing (SEC) is to deploy computing and storage resources to Low Earth Orbit (LEO) satellites at the edge of space-ground network, so that the on-orbit computing capabilities can be used to achieve online task processing[2-4]. Therefore, through the cooperation of LEO satellites and ground data centers, it is possible to provide users with multi-level, low-latency, and global coverage computing services.

However, in contrast with the edge computing in the terrestrial network, the service pattern of on-orbit satellite computing has brought new challenges. There are two major characteristics of resource limitation and dynamic in SEC. On the one hand, due to the limited cost of satellite, the payload has the requirements of low energy consumption and small size. So that, only various embedded boards with simplified systems, smaller volumes, and lower energy consumption can be used for computing support [5-7]. On the other hand, because the satellites are in uninterrupted orbital motion, the computing platform on board is in real-time motion relative to the ground, that is, the service area on the ground changes with time. Coupled with the dynamic changes of the inter-satellite network topology, SEC has the dual dynamic characteristics of network and service.

So as to improve the service capability of on-orbit computing resources, a resource management solution for satellite dynamics is an important research direction. For service dynamics, there are

significant differences in business characteristics in various regions around the world. For example, in the polar and ocean region, the business traffic is small and the type is mostly remote sensing image processing. In the bustling urban area, the business traffic is more intensive and mostly for time-sensitive tasks. Hence, under the condition of different regionalization of services, the satellite needs to optimize the configuration of existing resources in real-time during the movement process.

However, as the SEC research is in the initial stage [8], there is no relevant research to optimize the dynamic computing resources management of satellites. And it is unreasonable to apply ground-based research directly to the satellite computing platform. In this paper, according to the special space computing environment, we proposed a dynamic predictive resource management strategy for SEC, so as to reduce energy consumption of satellite computing platform while meeting the business needs of different regions. Specifically, the proposed Satellite Resources Two-step Management Strategy (SRTMS) utilizes a resource predictive strategy and a real configuration policy to dynamically scale the number of VMs, so as to optimize energy consumption of computing resource on LEO satellite constellation.

The content of this paper is organized as follows: The section 2 describes the existing research work. System model is presented in section 3. The dynamic predictive VM deployment strategy is proposed in section 4. Section 5 presents the evaluation results. The last section is the conclusion.

2 RELATED WORK

As mentioned above, there is a lack of research on the management of satellite computing resources. Therefore, in this section, studies on the problem of dynamic virtual machines (VMs) management problem in ground are discussed. In the research of computing resource management aimed at reducing energy consumption, virtual resources are mainly managed, and the main idea is to reduce the number of running VMs. Sun et al. [9] proposed an efficient VM management method in mobile edge computing (MEC). Through the proposed ILP and Tetris algorithms, task requests are placed on the MEC servers that met the least demand to minimize the number of active MEC servers and reduced the 22.2% energy consumption. Zhang et al. [10] proposed a workflow-based heuristic algorithm AMS, which is oriented to applications with multi-level services and believes that VMs can be shared by multiple IoT devices. With the ability to determine the number and location of each type of service VM, AMS can reduce the number of running VM by 28.4% and the cost of deployment by 33.9% in cases where user satisfaction rates are comparable.

Several studies have addressed dynamic VM management by predicting the task demand to optimize energy consumption. Research [11] proposed the non-homogeneous Markov model to predict the resource demand of the data source, and uses the heuristic algorithm to solve the mapping between VM and server, so as to minimize the total task consumption cost. In the literature [12-14], they proposed to predict the future resource utilization based on the historical resources data to better manage the VM placement while maintaining the QoS guarantee. Based on the prediction of grey Markov model[12] ,the regression model [13] and the Markov chains [14], these studies reduce the data center energy consumption by more than 30%. Chehelgerdi-Samani et al. [15] proposed a framework called PCVM.ARIMA, which aims to minimize the number of physical machines. By dynamically integrating the VM, detecting physical machine overloads, and minimizing the SLA of the ARIMA prediction model, the framework achieves a significant reduction in energy consumption while increasing the QoS factor.

In the above ground-based research of VM, the location of server and service area is fixed, and the received traffic has periodic and constant characteristics in time and space. For example, in study of Tang et al. [16], the upper limit of traffic is predicted by analyzing the traffic data of real operators in typical regions, and the overall VNF placing is planned with the goal of minimizing the source required by providing services for all traffic. However, in the SEC scenario, there is a lack of a large amount of real data, and the service area changes with the orbit and the rotation of the earth. Hence, the business traffic can only be predicted through the periodic orbital movement of the satellite, which cannot be applied to the spatio-temporal characteristics of the ground.

3 SYSTEM MODEL

3.1 Ground Business Model

The global region is divided into A set of non-overlapping subdomains $\mathcal{A} = \{1, ..., A\}$. For any region $a \in \boldsymbol{\mathcal{A}}$, there is a weight as follows:

$$W_a = WS_a + \alpha WP_a \tag{1}$$

$WS_a$is the static weight, which is determined by the basic flow of regional business (which can generally be estimated by regional population density) through the max-min standardized formula. The WP_arepresents the priority weight, which is manually set to affect the total weight of the region. The α is an impact factor.

In any region a, there is a random set of users $U_a = \{u_a^1, u_a^2, \ldots, u_a^{n_a}\}$ and a list of business types $List_a$. The n_a is the total number of users in the region, which is positively correlated with WP_a. And $List_a$ is a static table set by the regional business characteristics. The user u_a^i generates a task request $Task_a^i = \{\gamma_i, type_i, d_i\}$ randomly, where γ_i represents the resource amount of the task request and $type_i$ represents the task type of the request, with $type_i \in List_a$. The d_i represents the tolerable delay of the task, and the task whose result is not returned before this time is considered as a failure.

3.2 Satellite Resource Model

For the satellite cluster, it can be represented as a set $S = \{s_1, s_2, ..., s_n\}$, and each satellite carries heterogeneous computing resources for on-orbit task processing. For any satellites$_i$ = $\{orb_i, cover_i, rsc_i\}$, orb_i and $cover_i$denote the orbital attribute and the coverage attribute of the satellite payload. The orb_i includes six orbital elements (semi-major axis, eccentricity, orbital inclination, right ascension of ascending node, argument of perigee, and true anomaly), represented as $orb_i = \{a, e, i, \Omega, \omega, tp\}$. For the coverage attribute $cover_i = \{\psi, (\lambda, \varphi)\}$, ψ is the coverage angle, which can be calculated through the minimum observation angle θ. (λ, φ) is the longitude and latitude coordinate set of sub-satellite points, including $(\lambda, \varphi) = \{(\lambda, \varphi)^1, ..., (\lambda, \varphi)^T\}$ when the satellite service time is discretized into an equal time slot representation $\boldsymbol{\mathcal{T}} = \{1, ..., T\}$.

$rsc_i = \{C_i, C_i^t\}$ is the attribute of computing resources, where C_i represents the total amount of single satellite computing resources. In order to simplify the model, it is assumed that computing tasks are performed by VMs relying on physical computing resources, and all VMs use similar processing power C_{vm} and energy consumption $E_{vm}$when running. Therefore, the total amount of resources $C_i$can be represented by the maximum number of VM that can be run. Same as above, C_i^t represents the amount of remaining resources in slot t, which is the number of VMs that are not running but available. Furthermore, there is a set of running VMs $V_{s_i} = \{v_{s_i}^1, v_{s_i}^2, ..., v_{s_i}^{C_i}\}$on the satellite s_i. For a single running VM $v_{s_i}^j$, the available processing power of in time slot t is $C^t(v_{s_i}^j)$.

4 VM RESOURCE DYNAMIC SCALING STRATEGY

In order to reduce the unnecessary energy consumption of on-orbit computing resources, we propose a VM dynamic scaling management strategy based on traffic prediction. The main idea of the strategy is to combine the characteristics of orbital motion and ground region, and to dynamically scale the number of VMs by predicting the traffic in the next region to be served by the satellite. Through that, it can reduce the number of active VMs in the region with low traffic, so as to reduce energy consumption.

4.1 Dynamic Predictive Strategy for VM Instances using Historical Deviation Values

First of all, it is necessary to forecast the business traffic of the future region. There are two important points to note here:

1. To provide high availability of services, we should always ensure that there are sufficient resources to service most the tasks received on the satellite, so the predicted parameters should meet the upper limit of business traffic.

2. To simplify calculations and adapt to the out-of-sync feature of the satellites in the constellation, each satellite is responsible for its own VM management and traffic prediction for the next service region in the time slot t.

The subset of regional order of the satellite s_i path is defined as $\mathcal{A}_{s_i} = \{A_{s_i}{}^1, A_{s_i}{}^2, ..., A_{s_i}{}^n\}$. In each region, the service time is $T(A_{(s_i)}^j) = k\Delta T$, where k is a positive integer.

Therefore, the traffic prediction strategy is defined as follows: When the satellite $s_i$enters region $A_{s_i}{}^{j-1}$, the number of VMs in the region $A_{s_i}{}^j$ is predicted. The predicted number of required VMs is defined as:

$$B(A_{s_i}^j) = \begin{cases} \frac{W_{A_{s_i}^j}}{W_{A_{s_i}^{j-1}}} \times G(A_{s_i}^{j-1})(1+\lambda f[A_{s_i}^{j-2}]), 1 < j \le n \\ \beta W_{A_{s_i}^j}, j = 1 \end{cases} \tag{2}$$

The $B(A_{s_i}^j)$ is the minimum number of VMs meeting the future maximum traffic of region $A_{s_i}{}^j$. The number of VMs in the initial region is linearly correlated with the current region weight, and β is the influence factor. The number of VMs in the latter-order region is positively correlated with the weight ratio of the previous region. At the same time, in order to ensure the principle of high service availability and to modify the prediction deviation of historical areas, incremental service factor $\lambda f[A_{s_i}^{j-2}]$ is added to the demand prediction formula. λ is an incremental impact parameter between 0 and 1, representing the importance of historical business data to the prediction.

$f[A_{s_i}^{j-2}]$ is defined as the deviation adjustment formula based on the task failure rate in the historical region. Here, we introduce the adjustment principle $3-\sigma$[16], that is, the increased proportion of VM instances according to the historical failure rate is $\overline{e}(A_{s_i}^{j-2}) + 3\sigma(A_{s_i}^{j-2})$. The $\overline{e}(A_{s_i}^{j-2})$ is the average failure rate of $A_{s_i}^{j-2}$ and previous regional tasks, and $\sigma(A_{s_i}^{j-2})$ is the standard deviation of failure rate. The average failure rate[16] is defined as:

$$\overline{e}(A_{s_i}^{j-2}) = \frac{1-\mu}{1-\mu^N} \sum_{J=j-2-N}^{j-2} \mu^{j-2-J} e(A_{s_i}^{j-2-N}) \tag{3}$$

where$\mu \in [0, 1]$ represents the impact of historical data on the average failure rate, which is generally set to close to 0, that is, the farther away the historical area is from the current area, the less impact the failure rate has on the current adjustment. Correspondingly, the mean variance of failure rate is defined as:

$$v(A_{s_i}^{j-2}) = \frac{1-\mu}{1-\mu^N} \sum_{J=j-2-N}^{j-2} \mu^{j-2-J} [e(A_{s_i}^{j-2-N}) - \overline{e}(A_{s_i}^{j-2-N})]^2 \tag{4}$$

The standard deviation is:

$$\sigma(A_{s_i}^{j-2}) = \sqrt{v(\sigma(A_{s_i}^{j-2}))} \tag{5}$$

Based on the above definition, the number of incremental VM instances is defined as:

$$\frac{W_{A_{s_i}^j}}{W_{A_{s_i}^{j-1}}} \times G(A_{s_i}^{j-1}) \times \lambda(\overline{e}(A_{s_i}^{j-2}) + 3\sigma(A_{s_i}^{j-2})) \tag{6}$$

The details of the $G(A_{s_i}^{j-1})$ function is defined in the formula (8). As for $B(A_{s_i}^j)$, the predicted number of VMs in each region, will not exceed the maximum number of supported instances of on-orbit computing resources, which is expressed as follows:

$$\begin{aligned} &B(A_{s_i}^j) \le C_i \forall \mathrm{i}, \mathrm{j} \\ &B(A_{s_i}^j) \in \mathrm{Naturalnumberset}N \end{aligned} \tag{7}$$

4.2 VM Configuration Policy Based on Thresholds

After calculating the predicted number of VMs, it is necessary to combine the actual task process with the concept of threshold to determine the actual number of VMs to be maintained. The received task process needs to be explained: when the communication beam of satellite s_i covers the region $A_{s_i}{}^j$, users randomly distributed in the region will initiate task requests to the satellite. The task will be processed by the local VM after the satellite accepts it, and the Run-to-completion mode will be used for sequential calculation.

Since the startup and shutdown of VM instances have corresponding losses in terms of time and energy, we need to limit unnecessary changes in the scale of VMs. Therefore, the actual number of VMs $G(A_{s_i}^j)$ is introduced here, which can be defined as:

$$G(A_{s_i}^j) = \begin{cases} G(A_{s_i}^{j-1}), 1-\mathrm{R}_e \le \frac{B(A_{s_i}^j)}{B(A_{s_i}^{j-1})} \le 1 + R_e \\ B(A_{s_i}^j), \mathrm{other} \end{cases} \tag{8}$$

R_e is the scaling capacity threshold, and the optimal value of it can be determined by a large number of data experiments. The initial setting here is 0.1. Specifically, when the predicted number changes are greater than or less than the threshold, the actual number will be determined according to the predicted number. Otherwise, the number of VMs does not change if the threshold is not reached.

The actual operation of VM configuration needs to be divided into two parts: scaling up and scaling down. Existing studies [16] investigated the time consumption of different VM operations based on the proprietary test platform of real operator DCN. The average time to create a new VM is about 6 minutes, and it takes 5 seconds to shut down a VM. Therefore, when the predicted business traffic is smaller, we can scale down the scale of VMs to reduce the energy consumption of physical computing resources.

Hence, we can derive the optimal time for the actual operation of VM configuration as:

(1) VM scaling up: Since it takes a long time to create a VM, the scaling up operation will execute when it enters the region $A_{s_i}^{j-1}$ and after $G(A_{s_i}^{j})$ is determined.

(2) VM scaling down: Since it takes a short time to shut down a VM ,the scaling down operation executes after leaving the region $A_{s_i}^{j-1}$.

4.3 The Two-step Integrated Management Strategy SRTMS

In this section, a two-step integrated management strategy SRTMS is proposed based on the mentioned strategies. The integrated strategy is independently operated by the satellite, which is divided into two stages: prediction and actual configuration. When the satellite enters a new region ${A_{s_i}}^{j-1}$, the minimum number of VMs satisfying the future maximum traffic of region ${A_{s_i}}^{j}$ is firstly calculated through the prediction strategy, that is, the theoretical value of $B(A_{s_i}^{j})$. Next, the VM configuration strategy based on threshold value is used to calculate the actual number of instances which is the $G(A_{s_i}^{j})$. By implementing the two-step strategy on all the satellites in the satellite constellation, the VMs can be scaled in real-time and the total energy consumption of constellation can be reduced.

For the satellite s_i, the list $\Lambda[\boldsymbol{\mathcal{A}}_{s_i}]$records the actual number of VMs $G(A_{s_i}^{j-k})$, the number of theoretical VMs $B(A_{s_i}^{j-k})$, the area weight W_{Ak} and task failure rate $e(A_{s_i}^{j-k})$for each region that has passed through. The algorithm pseudo-code is as follows:

Algorithm 1 SRTMS Algorithm

Input: List of results for completed regions$\Lambda[\boldsymbol{\mathcal{A}}_{s_i}]$and the weight of the next area W_A.
Output: The result list of the area$\Lambda[\boldsymbol{\mathcal{A}}_{s_i}]$, and the actual number of instances in the next area $G(A_{s_i}^{j})$.
Procedure:
1: **while** $\Lambda[\boldsymbol{\mathcal{A}}_{s_i}]$ list has values **do**
2: Read $e(A_{s_i}^{j-k})$ according to the backtracking range N(set initially at 3).
3: Calculate the average historical failure rate $e(A_{s_i}^{j-2})$and standard deviation $\sigma(A_{s_i}^{j-2})$ according to formulas (3) and (5) .
4: **end while**
5: Obtain the deviation adjustment result $f[A_{s_i}^{j-2}]$.
6: Through formula (2) and the current regional weight W_{A-1}, the next regional weight W_A and the number of current regional instances$G(A_{s_i}^{j-1})$, the number of theoretical VMs of the next region $B(A_{s_i}^{j})$ is calculated.
7: Calculate actual number of instances $G(A_{s_i}^{j})$ by formula (7) and constraint (8).
8: Update$\Lambda[\boldsymbol{\mathcal{A}}_{s_i}]$
9: **return** $\Lambda[\boldsymbol{\mathcal{A}}_{s_i}]$ and $G(A_{s_i}^{j})$.

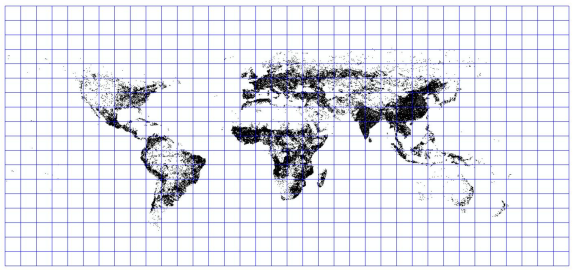

Figure 1: LandScan Global Population Data Grid.

5 EVALUATION RESULTS

5.1 Experimental Setup

The satellite-ground communication simulation is mainly based on STK (Satellite Tool Kit) software, and the dynamic configuration simulation of VM resources is mainly implemented by MATLAB. The satellite-ground regional visibility is provided by the STK simulation results, and the regional static weight data is calculated by the LandScan global population dataset [17] developed by the Oak Ridge National Laboratory (ORNL) of the US Department of Energy.Based on the LandScan, the global regional weight data grid that divides regions using equal latitude and longitude is shown in the Figure 1

5.1.1 The Constellation Parameters. In STK, Walker constellation is simulated: the orbital altitude is 1000 km, there are 3 orbital planes, the orbital inclination is 45 degrees, and each orbital planes 8 satellites. Therefore, there are a total of 24 satellites. Figure 2 shows the dynamic changes result of satellite-ground communication. The curves are the orbits of the satellites, and the circles are the satellite-ground communication regions.

5.1.2 Random Task Generator. Implemented by MATLAB, random generation of non-uniform user tasks in different areas is realized. Input parameters are shown in the Table 1

5.1.3 VM Instance-related Parameters. We set the total amount of resources $C_i$on a single satellite as follows: The number of CPUs is 8, and the total amount of memory is 8 GB. We set the total memory of a single VM to be 1GB, the number of CPU of a single VM to be 1, and the power consumption of a single VM to be 20 W.

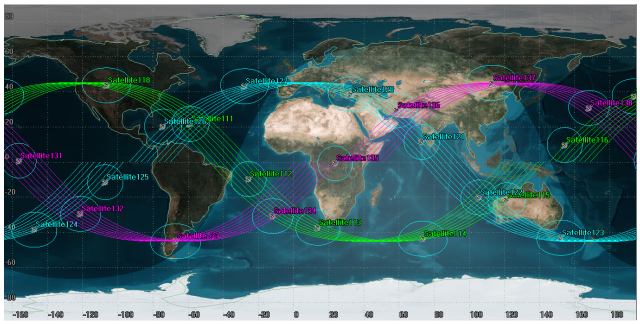

Figure 2: Dynamic Changes of Satellite-ground Communication.

Table 1: Task Generator Input Parameter

Maximum number of users in each area	n=10,20,30,40,50,60,70,80,90,100
Biggest task request resources	$\gamma_i = 100Mbit$(memory), 30s(CPU)
Task request types:	$\lvert List_a \rvert = 1$
Maximum task delay	$d_i = 5s$

Fifty experiments were conducted under each user number value, and the user task distribution was recorded in each experiment.

5.2 Results

Based on the aforementioned parameters, the experiments compare the performance of three instance modes in full capacity VM, half capacity VM, and SRTMS in terms of energy consumption.

The total energy consumption is set using:

$$E = \sum \sum G(A_{s_i}^j) \times T(A_{s_i}^j) \times E_{vm} \tag{9}$$

Figure 3 shows the changes in total VM energy consumption of a single satellite with the maximum number of users in the region within 24 hours. Figure 4 shows the changes in the average failure rate of a single satellite with the maximum number of users in the region within 24 hours. The average mission failure rate is defined as the average ratio of failed tasks to the total number of tasks in different regions. Task failures occur because of missing deadlines or running out of resources.

In the full capacity mode, the number of single satellite VMs always keeps the maximum running number, that is, the situation that all resources are occupied in the traditional satellite computing mode. In half capacity mode, the number of VMs should always be half of the maximum number, and the energy consumption should be in half value of full capacity mode. Therefore, in full capacity mode and half capacity mode, the number of VMs and energy consumption remains constant and does not change with the number of users.

For the entire constellation within 24 hours, the total energy consumption and mission failure rate when the maximum number of users is set to 80 is shown in Table 2

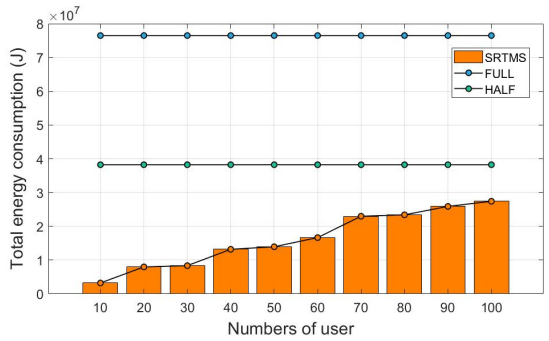

Figure 3: The Variation of Total Energy Consumption of VM of a Single Satellite Varies with the Maximum Number of Users in the Region Within 24 Hours.

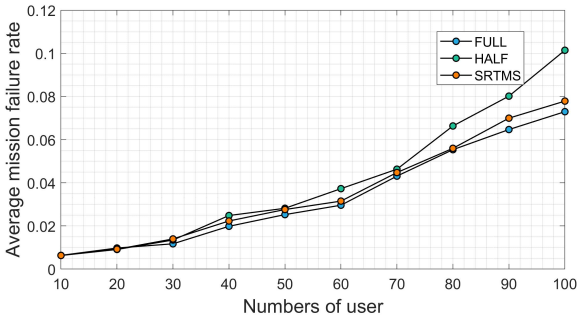

Figure 4: The Variation of the Average Mission Failure Rate of a Single Satellite in 24 Hours varies with the Maximum Number of Users in the Region.

Table 2: Total Energy Consumption and Mission Failure Rate of VM in 24 Hours across the Constellation

Mean	Average mission failure rate	Total energy consumption of VM (J)
Full VMs	5.23%	$2.14*10^9$
Half VMs	8.61%	$1.07*10^9$
SRTMS	7.37%	$5.73*10^8$

It can be seen that the SRTMS algorithm reduces the energy consumption by 73% compared to the full capacity VM mode and 46% compared to the half capacity VM mode.

From the above figures and table, it can be seen that if the task failure rate does not significantly change, the optimization of total energy consumption in the SRTMS algorithm will remain at a good level. On the other hand, due to the continual change of the number of VMs in SRTMS, the capacity may not be scaled up in time in some high-traffic regions. So the failure rate performance is worse than that of the full capacity mode. In future work, the failure rate of SRTMS can be reduced by optimizing the scaling threshold and the historical backtracking range.

6 CONCLUSIONS

In this paper, based on the special SEC scenario, we have introduced a dynamic predictive two-step VM scaling strategy SRTMS, which aims at reducing redundant energy consumption of satellite computing platform while meeting the business requirements of different regions. The evaluation results show that SRTMS method can obtain an effective solution, scaling the number of VMs for the dynamic ground region in time. Under the condition that the task requirements are basically satisfied, the total energy consumption of on-orbit VMs is greatly reduced, which is 73% lower than the baseline of always running all VM resources.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (2020YFB1806000).

REFERENCES

[1] W-C Chien, C-F Lai, M S Hossain, *et al.* (2019). Heterogeneous Space and Terrestrial Integrated Networks for IoT: Architecture and Challenges. Ieee Network, 33, 15-21.

[2] Z J Zhang, W Y Zhang and F H Tseng (2019). Satellite Mobile Edge Computing: Improving QoS of High-Speed Satellite-Terrestrial Networks Using Edge Computing Techniques. Ieee Network, 33, 70-76.

[3] S Liao, M Dong, K Ota, *et al.* (2018). Vehicle Mobility-Based Geographical Migration of Fog Resource for Satellite-Enabled Smart Cities. 2018 IEEE Global Communications Conference (GLOBECOM), pp 1-6.

[4] P Wang, J Zhang, X Zhang, *et al.* (2018). Performance Evaluation of Double-edge Satellite Terrestrial Networks on OPNET Platform. 2018 IEEE/CIC International Conference on Communications in China (ICCC Workshops), pp 37-42.

[5] P L Dai, Z H Hang, K Liu, *et al.* (2020). Multi-Armed Bandit Learning for Computation-Intensive Services in MEC-Empowered Vehicular Networks. Ieee Transactions on Vehicular Technology, 69, 7821-7834.

[6] N Slamnik-Krijestorac, E D E Silva, E Municio, *et al.* (2020). Network Service and Resource Orchestration: A Feature and Performance Analysis within the MEC-Enhanced Vehicular Network Context. Sensors, 20, 28.

[7] K Peng, B Zhao, S Xue, *et al.* (2020). Energy- and Resource-Aware Computation Offloading for Complex Tasks in Edge Environment. Complexity, 2020, 1-14.

[8] R C Xie, Q Q Tang, Q N Wang, *et al.* (2020). Satellite-Terrestrial Integrated Edge Computing Networks: Architecture, Challenges, and Open Issues. Ieee Network, 34, 224-231.

[9] Y Sun, X Chen, D Liu, *et al.* (2019). Power-Aware Virtual Machine Placement for Mobile Edge Computing. 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp 595-600.

[10] X Zhang, H Huang, H Yin, *et al.* (2019). Resource Provisioning in the Edge for IoT Applications With Multilevel Services. IEEE Internet of Things Journal, 6, 4262-4271.

[11] K Xiao, Z Gao, Q Wang, *et al.* (2018). A Heuristic Algorithm Based on Resource Requirements Forecasting for Server Placement in Edge Computing. 2018 IEEE/ACM Symposium on Edge Computing (SEC), pp 354-355.

[12] S Y Hsieh, C S Liu, R Buyya, *et al.* (2020). Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers. Journal of Parallel and Distributed Computing, 139, 99-109.

[13] K Haghshenas and S Mohammadi (2020). Prediction-based underutilized and destination host selection approaches for energy-efficient dynamic VM consolidation in data centers. Journal of Supercomputing, 76, 10240-10257.

[14] A Tarafdar, M Debnath, S Khatua, *et al.* (2020). Energy and quality of service-aware virtual machine consolidation in a cloud data center. Journal of Supercomputing, 76, 9095-9126.

[15] M Chehelgerdi-Samani and F Safi-Esfahani (2020). PCVM.ARIMA: predictive consolidation of virtual machines applying ARIMA method. The Journal of Supercomputing.

[16] H Tang, Zhou and D Chen (2019). Dynamic Network Function Instance Scaling Based on Traffic Forecasting and VNF Placement in Operator Data Centers. IEEE Transactions on Parallel and Distributed Systems, 30, 530-543.

[17] A N Rose, J J McKee, M L Urban, *et al.* (2019). LandScan 2018. 2018 ed., Oak Ridge National Laboratory, Oak Ridge, TN.