Computers, Materials & Continua CMC, vol.61, no.1, pp.423-437, 2019

A Heterogeneous Virtual Machines Resource Allocation Scheme in Slices Architecture of 5G Edge Datacenter

Changming Zhao[1, 2, *], Tiejun Wang[2] and Alan Yang[3]

Abstract: In the paper, we investigate the heterogeneous resource allocation scheme for virtual machines with slicing technology in the 5G/B5G edge computing environment. In general, the different slices for different task scenarios exist in the same edge layer synchronously. A lot of researches reveal that the virtual machines of different slices indicate strong heterogeneity with different reserved resource granularity. In the condition, the allocation process is a NP hard problem and difficult for the actual demand of the tasks in the strongly heterogeneous environment. Based on the slicing and container concept, we propose the resource allocation scheme named Two-Dimension allocation and correlation placement Scheme (TDACP). The scheme divides the resource allocation and management work into three stages in this paper: In the first stage, it designs reasonably strategy to allocate resources to different task slices according to demand. In the second stage, it establishes an equivalent relationship between the virtual machine reserved resource capacity and the Service-Level Agreement (SLA) of the virtual machine in different slices. In the third stage, it designs a placement optimization strategy to schedule the equivalent virtual machines in the physical servers. Thus, it is able to establish a virtual machine placement strategy with high resource utilization efficiency and low time cost. The simulation results indicate that the proposed scheme is able to suppress the problem of uneven resource allocation which is caused by the pure preemptive scheduling strategy. It adjusts the number of equivalent virtual machines based on the SLA range of system parameter, and reduces the SLA probability of physical servers effectively based on resource utilization time sampling series linear. The scheme is able to guarantee resource allocation and management work orderly and efficiently in the edge datacenter slices.

.

Keywords: Heterogeneous virtual machine, resource allocation, edge computing, slicing.

1 Introduction

In the new generation mobile communication system (5G) technology architecture, it is usually separating and offloading the computation-intensive and the energy-intensive segments of running tasks to the edge datacenter in the ultra-dense network [He, Ren, Yu

[1] College of Computer Science, Sichuan University, Chengdu, 610041, China.

[2] School of Computer Science, Chengdu University of Information Technology, Chengdu, 610225, China.

[3] Amphenol AssembleTech, Houston, TX 77070, US.

[*] Corresponding Author: Changming Zhao. Email: zcm84@cuit.edu.cn.

et al. (2019)]. Therefore, the resource allocation and management scheme become the key problem to limit the operation efficiency in the edge datacenter virtualization. The conventional virtual resource allocation and management strategies probably results in resource mismatch in the environment of multi-dimensional resources allocation independently. Network slicing is a new task-oriented concept introduced in 5G/B5G technology architecture for the domain of resource allocation and management [Ning, Wang, Huang et al. (2019)]. In the concept, it aims to provide dedicated, isolated and reliable services to establish an integrated computing and communication resources at the task granularity. Therefore, it provides a new technical methodology for the virtual resource allocation and management in the edge datacenter [Cui, Gong, Ni et al. (2019); IMT 2020 (5G) Promotion Group (2014); Hu, Patel and Sabella (2018)].

Basically, the novel task slicing technology possesses three main technical characteristics, such as virtualization, specialization and isolation. The virtualization refers that the task slices system is necessary to establish on the NFV/SDN infrastructure [Ma, Wen, Wang et al. (2018)]. The specialization is implying that each slice is tailored for different service demands. Each slice is assigned for sufficient resources as virtual computing, network bandwidth and service quality [Richart, Baliosian, Serrat et al. (2016); Wen, Feng, Tang et al. (2019); Xiong, Leng, Hu et al. (2019)]. The isolation means that arbitrary slices is independently in resource allocation and management operation [Sun, Peng, Mao et al. (2019); Rost, Breitbach, Roreger et al. (2018); Vo, Nguyen, Le et al. (2018)]. Therefore, any probable failure in one slice shall be constricted inside itself and less impact for the other slices.

The traditional global resource allocation strategy is operating based on virtual machine granularity. In these allocation strategies, they set several classes of constant virtual machine resource granularities for allocation according to historical statistics and maps the entire tasks to the most appropriate matched virtual machine granularities. Abdullahi researches the allocation scheme with the parameter of virtual machine minimization completing time based on the Service-Level Agreement (SLA) levels [Abdullahi, Ngadi and Dishing (2017)]. Garg designs a min-min compromise algorithm to minimize the weighted sum of utilization resources and execution time [Garg, Buyya and Siegel (2010)]. Burger studies the virtual machine resource allocation scheme in heterogeneous environments that try to match the dedicated resources virtual machines with offloading tasks in a heterogeneous environment [Burge, Ranganathan and Wiener (2007)]. These resource allocation strategies contain the advantage of low execution time cost. And the task resource allocation is independent for the entire virtual machines. In view of these, the resources match with less effectiveness at task granularity for the traditional strategies. Therefore, it is necessary to reconstruct the edge domain data services system by task slices to make the allocation operation task-oriented and logically. But from a global perspective, the resources granularity of virtual machine indicates more heterogeneous where it is deployed in different slices. Based on above, the existing virtual machine resource allocation system is necessary to be improved.

On the other hand, after the virtual machine granularity allocation, the task scheduling is based on the static placement strategy. For example, Tiwari proposes a new scheduling strategy on Rough Sets Theory (RST) [Tiwari, Nagaraju and Mahrishi (2010)]. Shen

provides a novel resource rent scheduling strategy to prove the resource utilization efficiency by resource reserved and on-demand simultaneously [Shen, Deng and Iosup (2013)]. Breitgand formalizes the virtual machine scheduling process as a combination optimization problem [Breitgand, Kutiel and Raz (2010)]. In the optimization problem, it takes the SLA parameter of the task as the core variable to structure equations. On the basis of the equations, Breitgand proposes a strategy to solve the minimum resource demand with a given SLA threshold by system. The above strategies take into the SLA factor in the scheduling process. However, the strategies still use the static parameter to establish the model. It might to cause scheduling in inefficiency with high likelihood. Therefore, the scheduling strategy is also necessary to be improved.

In this paper, we propose a novel heterogeneous virtual machines resource allocation scheme based on the virtualization and slicing technologies in the 5G technology architecture. The scheme is named Two-Dimension allocation and correlation placement Scheme (TDACP) and able to divide into three stages. In the first stage, it designs reasonably strategy to allocate resources to different task slices on demand. The strategy is able to constating the resource share of strong demand slices. In the second stage, it establishes an equivalent relationship between the virtual machine reserved resource capacity and the SLA of the virtual machine in different slices. In the proposed equivalent equation, it transforms the original virtual machines to the equivalent virtual machines with the SLA demand. In the third stage, it designs a placement optimization strategy to schedule the equivalent virtual machines in the physical servers. Thus, it is able to establish a virtual machine placement strategy with high resource utilization efficiency and low time cost.

This paper includes five chapters. In the first chapter, we provide the introduce for the research background and motivation. We also present the research object of this paper in the chapter. In the second chapter, we provide the theoretical fundamental of this paper. In the chapter, we present the details of related theories and formulation models. In the third chapter, the section demonstrates the three strategies pseudocode corresponding to the three stages of propose scheme. In the fourth chapter, we design four numerical simulations to verify the three strategies in proposed TDACP scheme. In the fifth chapter, it is the summary and future works about our researches.

2 System model and theory analysis

In this chapter, we present two parts contents to research the theorical fundamental of the proposed scheme TDACP. The first part is the system model details and the illustration, and the second part is the theory analysis and the formulaic descriptions.

2.1 System model

In this paper, the system model frame is also able to divide into three stages. The frame of the paper is illustrated as the Fig. 1 in the follow. In the first and second stages, the system is summarized as a two-layer resource allocation process. And the role of third stage is to place the virtual machines into the physical servers of virtual edge data center. In addition, in order to simplify the description logic, the whole resource inside edge data

center is virtualized as one type of normalized resource (NMR). The unit NMR is generated as the resource ratio of the local edge data center. The proposed resource allocation frame is design for the NMR.

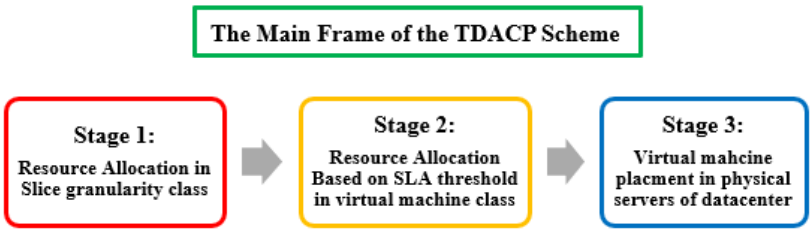

Figure 1: The main frame of the TDACP Scheme

In the first stage, the resource allocation is processing in the granularity of slice class. It defines the NMR into private NMR and share NMR. In the statistic period, arbitrary slice is banned from employing the private NMR of other slices. However, the slice is able to rob the slice share NMR once the slice depletes the private NMR itself. Complete this stage, the whole NMR is assigned to the slices.

In the second stage, the resource allocation proceeds in the granularity of virtual machine class. In the stage, the primary task is for the risk-resource model. This model assumes that the peak resource usage of the virtual machine satisfies the normal distribution. Based on the hypothesis, the probability distribution model between peak resource quantity and service quality can be established. The system sets several classes in the slice according as the different service priority. Therefore, the system possesses the ability to calculate the specific resource peak of arbitrary class in one slice corresponding to service priority. Based on this, we can obtain a certain overcommit virtual machines in an acceptable and relatively low SLA.

In the third stage, the system should place all virtual machines manufactured in the first two stages placement in the physical servers. During the placement process, system should monitor the resource utilization sampling time sequence of each virtual machine filling in same physical servers to avoid match the relative virtual machines, which may result in a peak superposition.

Through the resource allocation and scheduling in the three stages, the scheme is able to guarantee the datacenter running under efficiency condition in the 5G edge layer.

2.2 Theory analysis

In the first stage, when the terminal device connects with the system in the first time and requests resources from the system, the system could identify the type of task required of the device by the task plane of system. The task plane tests the correlation between the task and the whole existing standard network task slices. Once the task is able to match with one type of specific slice, all the class of devices should configure virtual machine resources with the standard of slice.

2.2.1 The analysis for slice class allocation

In the first stage, we design a strategy to complete the resource allocation in the slice class granularity. The process is illustrated in the Fig. 2.

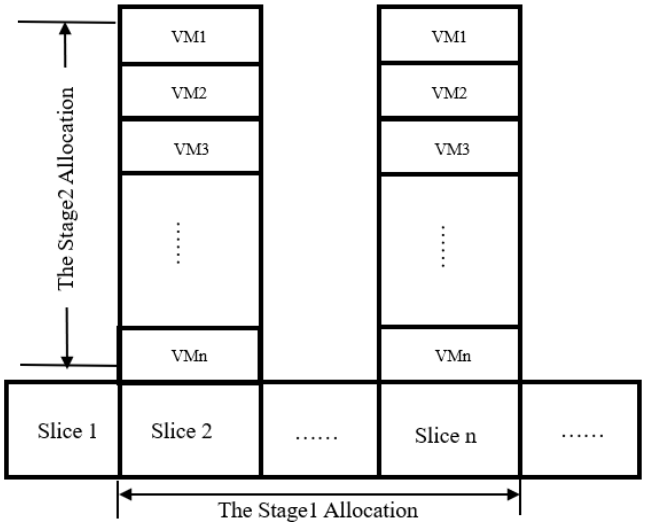

Figure 2: The two classes allocation plan

In the first stage, when the terminal device connects with the system in the first time and requests resources from the system, the system could identify the type of task required of the device by the task plane of system. The task plane tests the correlation between the task and the whole existing standard network task slices. Once the task is able to match with one type of specific slice, all the class of devices should configure virtual machine resources with the standard of slice.

We set the total amount of resources in the system as U_T, in which the private resource denoted as U_P and the share resource denoted as U_S. There is the following equation relative with the two parameters:

$$\begin{cases} U_T = U_P + U_S \\ \dfrac{U_P}{U_S} = r, \quad 0<r<1 \end{cases} \tag{1}$$

Let S_{total}^{j} represent the total resources occupied by all slices in the system at time j, and S_i^j represent the total resources occupied by slice i at time j.

$$\sum_{i=1}^{N} S_i^j = S_{total}^j \tag{2}$$

In general, S_{total}^{j} is not a constant, and it is defined as follow:

$$S_{total}^{j} \le \max(S_{total}^{j}, j \in T) \tag{3}$$

In general, the slice does not start to run with heavy load simultaneously, the amount of resources available for each slice is always a gradual increment process. When the total amount of resource occupied by slice i at moment j_1 is larger than the initial private resource amount for the first time, it starts to preempt resource from the shared resource U_S. For the resource amount $S_i^{j_1}$, the $[\![\frac{U_P}{N}]\!]$ stands for the resource occupied by slice i at moment j_1, and $\sigma_i^{j_1}$ means the resource application increment by slice i at moment.

$$S_i^{j_1} = [\![\frac{U_P}{N}]\!] + \sigma_i^{j_1} \tag{4}$$

However, in order to avoid few slices, occupy too much shared resource in a very short period. A slice should run the discount factor to modify its resource application value when its application accumulation over the threshold. Assume the slice i apply C_1 times share resource and appl C_2 times after its value larger than threshold, we form relationship between original application value σ_i^j and discount value $\langle \sigma_i^j \rangle$.

$$\langle \sigma_i^j \rangle = \sigma_i^j * (1 - \frac{C_2}{C_1 + C_2}) \tag{5}$$

When the system available resource is less than a given value, all the slice is banned to apply resource. The time point is denoted as j_2. And we can obtain:

$$\sigma_i^j = \begin{cases} \langle \sigma_i^j \rangle & , \ (1+r)U_S - \max(S_{total}^j, j \in T) > z \\ 0 & , \ (1+r)U_S - \max(S_{total}^j, j \in T) = z \end{cases} \tag{6}$$

for the slice i, the maximum applied resource after j_2 is $S_i^{j_2}$.

$$S_i^j \le S_i^{j_2} \ , \ j = j_2 \ , j_2 + 1 \ , j_2 + 2 \ , \ldots\ldots \ , \mathrm{T} \tag{7}$$

And we define new parameter as resource equivalent demand as:

$$\overline{S_i^j} = \frac{1}{Count(j_2, T)} * \sum_{j=j_2}^{T} S_i^j \tag{8}$$

After above operations, all the resource is distributed in the slices. In the second stage, it will allocate the resource inside each slice. In this stage, it is necessary to establish a resource priority class allocation model based on risk aware.

2.2.2 The analysis for Virtual Machine class allocation

In the second stage, we design a strategy to complete the resource allocation in the virtual machine class granularity. The process is illustrated in the Fig. 3.

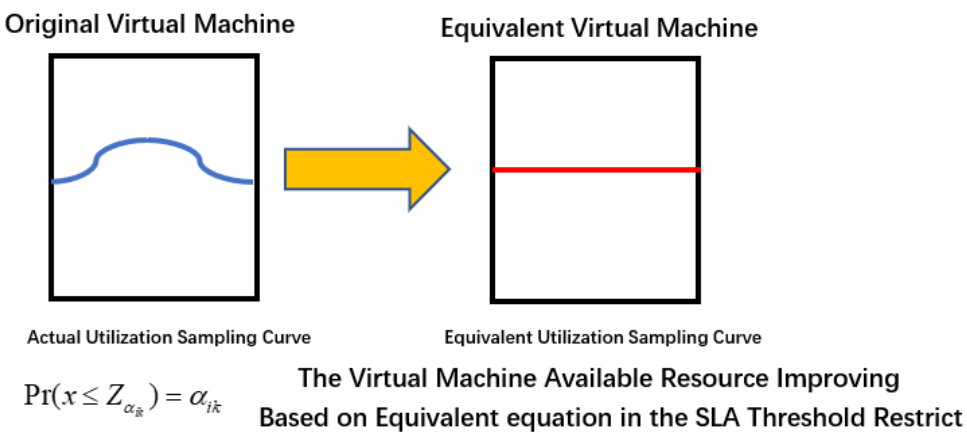

Figure 3: Equivalent virtual machine based SLA easing

In general, the same slice fits to the same service. And inside one slice, we also define sever different classes within different SLA levels. After the first stage of scheme, we suppose the slice i also records the peak demand of class k equivalent virtual machine , therefore the peak demand of class k equivalent virtual machine is:

$$\overline{v_{ik}^{j}} = \frac{1}{Count(j_2, T)} * \sum_{j=j_2}^{T} v_{ik}^{j} \tag{9}$$

By using the normal distribution model, the SLA of class k of slice i can be obtained：

$$\Pr(x \le Z_{\alpha_{ik}}) = \alpha_{ik} \tag{10}$$

If the SLA is available for the class k in the slice i, then

$$Ocm\_ratio = \frac{1}{Z_{\alpha_{ik}}} \tag{11}$$

for slice i, there exist $D2$ classes virtual machines, then the number of the overcommit virtual machines is:

$$Ocm\_Vnum\_slic(i) = \sum_{i=1}^{D2} m_k * (\frac{1+Z_{\alpha_{ik}}}{Z_{\alpha_{ik}}}) \tag{12}$$

We evaluate the risk of class k virtual machines in the slice i. Assume threshold of virtual machine number over the peak limit is 10%, and the rated number of virtual machines is $D1$. The risk probability of threshold is defined as:

$$Ocm\_risk\_slic(i)class(k) = \prod^{Random(\lceil D1*(1+1/Z_{\alpha_{ik}}) \rceil, 0.1)} (1-\alpha_{ik}) \tag{13}$$

For the slice, if there is no obvious correlation between the running virtual machines, the risk probability is acceptable. The risk is defined as:

$$Ocm\_risk\_slic(i) = \prod_{i=1}^{D2} \prod_{k=1}^{k=Random(\lceil D1*(1+1/Z_{\alpha_{ik}})\rceil,0.1)} (1-\alpha_{ik}) \tag{14}$$

2.2.3 The analysis for virtual machine correlation placement

In the third stage, we design a strategy to place the virtual machines in physical servers based on linear correlation theory. The concept is indicated as the Fig. 4.

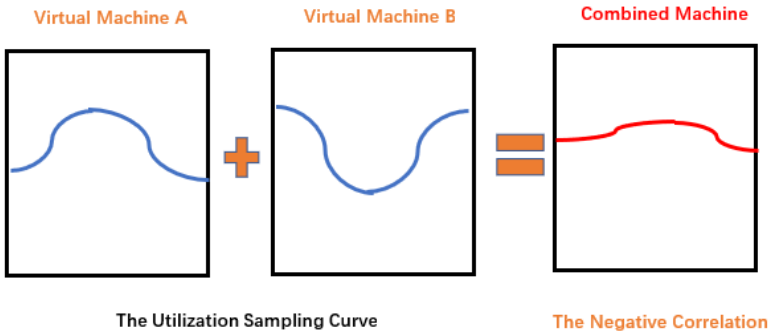

Figure 4: Placement based on linear correlation

In the third stage, we should place the virtual machines, which are generated in the first stage and the second stage, into the physical servers. And we introduce the linear correlation coefficient to test the correlation of two virtual machines. The first task is to establish the virtual machine time sampling model. Assume the virtual machine sampling frequency is 10 times than slice sampling. Then for the two virtual machines A and B, there are:

$$\begin{aligned} Sample\_v_A &= \{v_1^A, v_2^A, v_{3,.....,}^A v_{10T}^A\} \\ Sample\_v_B &= \{v_1^B, v_2^B, v_{3,.....,}^B v_{10T}^B\} \end{aligned} \tag{15}$$

And we make use of the time series sampling to establish correlation analysis model. If the correlation coefficient is negative, it is able to define the two virtual machines as a uncorrelated pair. The linear correlation coefficient is defined as:

$$Coeff(v_A, v_B) = \frac{\sum_{i=1}^{10T}(v_1^A - \frac{1}{10T}*\sum_{i=1}^{10T} v_i^A)(v_1^B - \frac{1}{10T}*\sum_{i=1}^{10T} v_i^B)}{\sqrt{\sum_{i=1}^{10T}(v_1^A - \frac{1}{10T}*\sum_{i=1}^{10T} v_i^A)^2}\sqrt{\sum_{i=1}^{10T}(v_1^B - \frac{1}{10T}*\sum_{i=1}^{10T} v_i^B)^2}} \tag{16}$$

In theory, the linear dependent strictly refers to:

$$Coeff(v_A, v_B) = 0 \tag{17}$$

However, in actual environment, we can relax the restrict as:

$$Coeff(v_A, v_B) \le |r| \tag{18}$$

It also to be defined as linear uncorrelated. Based on the Eq. (18), we proposed a new

virtual machine placement algorithm based on linear uncorrelated test.

The traditional uncorrelated placement algorithm is prone to fail in linear correlation coefficient after several rounds of consolidation. In this situation, it proposes a novel multi-round divide-and-conquer consolidation strategy in the paper. The strategy is described that making use of Eq. (16) to analyze the entire virtual machine in pairwise. If any virtual machine pair matches to the description in Eq. (18), the two-resource utilization time sampling serials are able to consolidate with each other by point to point superposition.

$$Peak(v_A) = \max(\{v_1^A, v_2^A, v_{3,\ldots\ldots,}^A v_{10T}^A\}) \tag{19}$$

We name the consolidation result as combined virtual machine. If the peak of the resource utilization serial of combined virtual machine is not over the given threshold described in the Eq. (19), the combined virtual machine is able to repeat the process to improve resource utilization. We provide the strategy details in the next chapter.

3 Strategy algorithm pseudocode

In this chapter, we provide the strategies algorithm pseudocodes according to the three stages in the scheme. The first algorithm pseudocode is about the slice class resource allocation in dynamic scene. The second algorithm pseudocode is about the virtual machine class resource allocation in static scene. The third algorithm pseudocode is about the virtual machine correlated placement in physical server in in static scene. The scheme combines the dynamic and static resource allocation models into one whole unit.

3.1 The fair allocation in slice class

The first part is the slice granularity class allocation strategy algorithm pseudocode. One slice is not able to allocate the share resource until the private resource of itself is allocation out. We use a variable weighted factor to constraint the virtual resource allocation as Eq. (6). The pseudocode is described as following:

Table 1: The slice class resource allocation algorithm

1.	David the entire resource U_S and U_P as Eq. (1), N= n;
2.	Each slice obtains a certain original private resource as Eq. (2);
3.	While ($U_T \notin \varnothing$)
4.	Do{
5.	While ($U_P / n \notin \varnothing$)
6.	Do{allocation $\sigma_i^{j_1}$ to $S_i^{j_1}$ for slice as Eq. (5) and Eq. (6) };
7.	While ($U_S \notin \varnothing$)
8.	Do{allocation $\sigma_i^{j_1}$ to $S_i^{j_1}$ for slice as Eq. (5) and Eq. (6) };
9.	}

3.2 The overcommit in virtual machine class

The second part is the virtual machine granularity class allocation algorithm pseudocode. In the algorithm, it is possible to overcommit allocate virtual machines resource in one slice by SLA easing. Basically, this is a feasible method that reduce the SLA demand for more available resource as Eq. (11) and Eq. (12). The pseudocode is described as following:

Table 2: The virtual machine class resource allocation algorithm

```
Calculate each class virtual machines as Eq. (9);
For (Slice class 1 to n)
    {
    For (Virtual machine class from 1 to m)
        {
            Run Eq. (10), Eq. (11) and Eq. (12);
        }
    }
}
Run Eq. (14).
```

3.3 The virtual machine correlated placement

The third part is the virtual machine correlated placement strategy algorithm pseudocode. In the algorithm, we make use of Eq. (16) and Eq. (18) to seek for the entire uncorrelated virtual machine pairs. The pseudocode is described as following:

Table 3: The virtual machine correlated placement algorithm

```
For (Virtual machine class from 1 to m)
    {
    For (Virtual machine class from 1 to m)
        {
            Calculate the entire virtual machines in pairs as Eq. (16);
            Judge the entire virtual machine pairs as Eq. (18);
        }
    }
Repeat placement process with uncorrelation virtual machines as Eq. (19).
```

4 Simulation result

In this chapter, we verify the effectiveness of the resource allocation framework proposed in this paper. The simulations are divided into three parts, corresponding to the three stages in this paper. The first part of the simulation is for slice level resource allocation. We assume that the normalized resource in the system needs to be allocated to four slices.

In order to verify the concept of private resources and discount coefficient proposed in the

first stage, two sets of simulations were set up for comparison. The first group used pure preemptive strategy for resource allocation, while the second group used the concept of private resources and discount coefficient. As shown in Fig. 5, the demand for resources from slice 1 to slice 3 at the beginning of the statistical period was too strong, and the system did not limit the application of resources for slices. When the slice 4 of resource demand starts up gradually at the end of the statistical period, the allocable resources have been exhausted. Slice 4 cannot apply enough resources, it can only run at a low level of resource allocation for a long time, which seriously affects the performance of the slice.

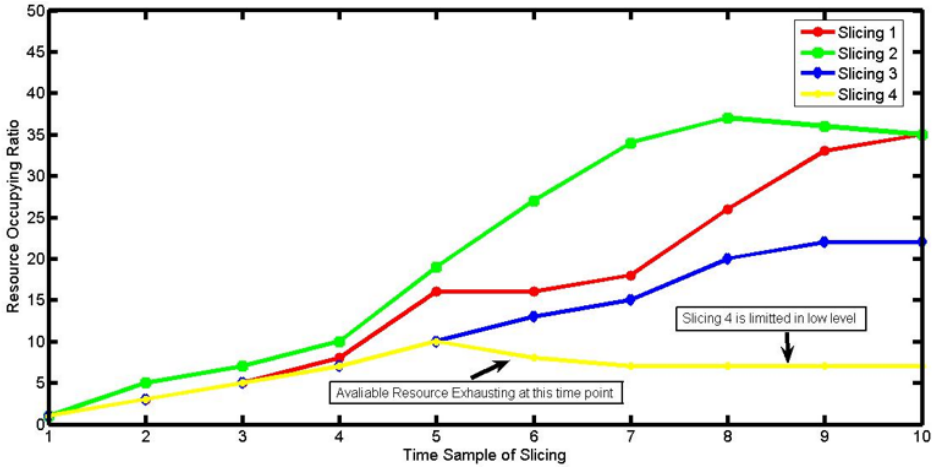

Figure 5: Pure preemptive resource allocation strategy for slice class

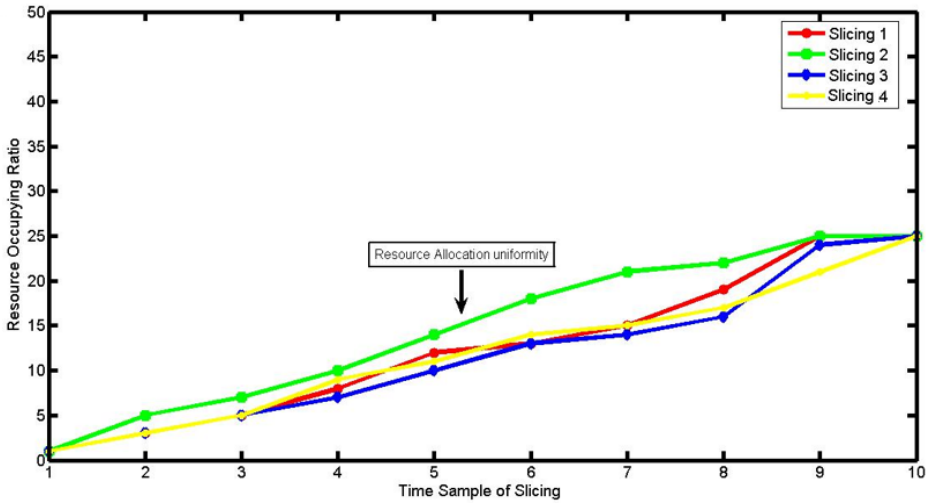

Figure 6: Mix strategy with discount factor

In Fig. 6, with the introduction of the concept of private resources and discount factor, the scheduling strategy possesses a good inhibition on the over-rapid rise of sliced resource applications. Therefore, in the middle of the statistical period, there is still a large amount

of free resources in the system. And the system automatically closes the resource application for the slice after the total resource for the slice over 25%. Therefore, all slices in this framework are able to allocate up to 25% resource only. The simulation results show that the framework proposed in this paper is effective in the first stage.

In the simulation of the second stage, the system established the model relationship between the virtual machine resource supply and SLA of any kind within the slice through statistical regression. Based on this model, the number of slice-borne virtual machines can be increased with the permission of the task SLA. Fig. 7 above is the virtual machine type managed by Section 3, which has 7 classes.

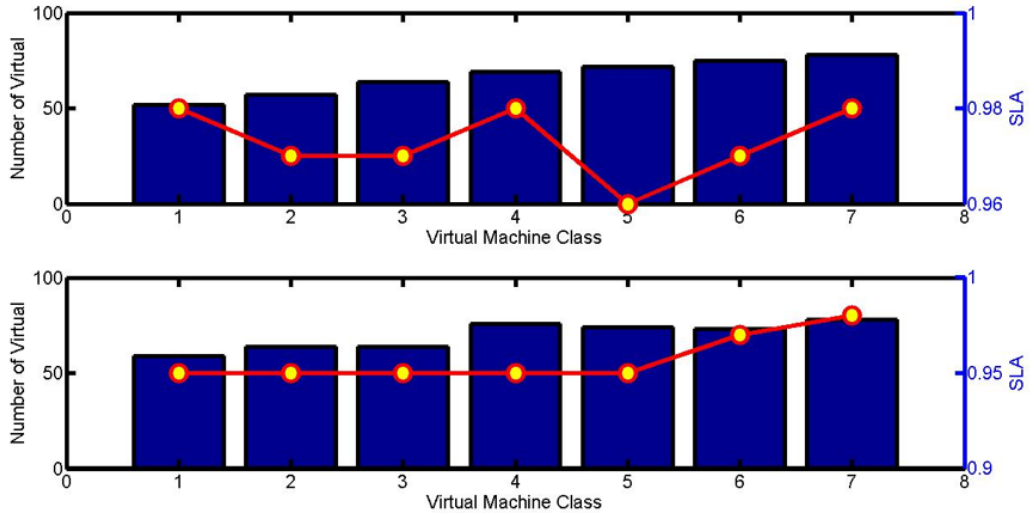

Figure 7: The relation between the number of virtual machine and SLA

The number of various virtual machines is distributed between 52 and 78, while the SLA is stable between 0.96 and 0.98. According to the requirements of service type, take Section 2 for example, where the SLA of classes 1 to 5 is able to relax to 0.95, and classes 6 and 7 need to be maintained at 0.98.

The Fig. 7 shows the number of various virtual machines and the measured SLA of various virtual machines. It can be seen that the change of measured SLA is highly consistent with the change of theoretically predicted SLA. Shows that the model relationship between the virtual machine source supply and the SLA in the second phase is established and available.

In the third stage, it is necessary to verify the placement strategy of the virtual machine in the resource allocation framework. There are 3,861 virtual machines in the initial state, requiring 78 physical servers to be fully hosted. At this point, it takes 77 servers to load at 0.1 and 74 physical servers to load at 0.2, and when it is 0.25, only 72 physical servers are required to load. But at 0.3, you need 75 physical servers to load. It is able to see that the virtual machine placement strategy proposed in this paper plays an obvious role in selecting appropriate correlation threshold. However, when the correlation threshold is not properly selected, there may be side effects. If the selection is too small, it may cause

too many uncorrelated virtual machines, which will make the final placement worse. On the contrary, once the selection is too large, the number of unaggregated virtual machines are increasing in the number of servers at the end.

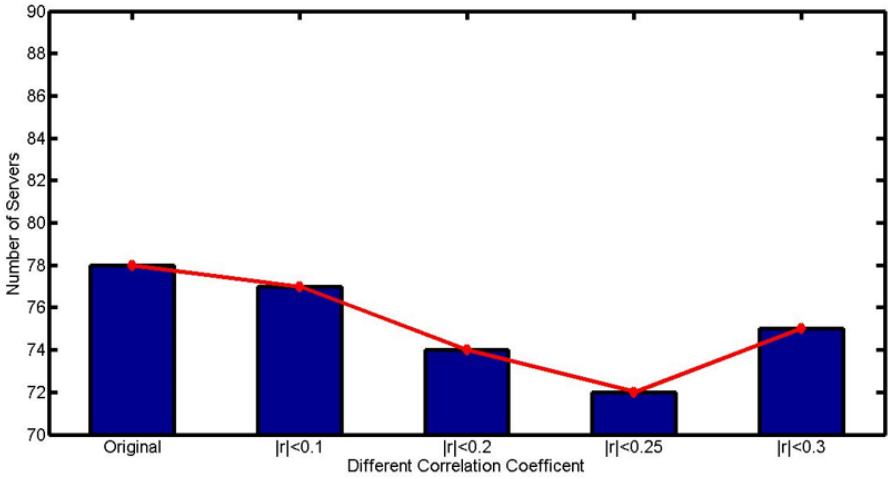

Figure 8: The result of different $|r|$

5 Summary and future work

In this paper, we propose a novel scheme，which is named Two-Dimension allocation and correlation placement Scheme (TDACP), based on the slices and vitalization container technologies. The theoretical analysis and numerical simulation indicate that the proposed scheme is able to suppress the problem of uneven resource allocation which is caused by the pure preemptive scheduling strategy. It is able to adjust the number of equivalent virtual machines based on the SLA range of system parameter. It is also able to reduce the SLA probability of physical servers effectively based on resource utilization time sampling series linear.

In the future, the TDACP scheme will be researched in the environment of Ultra-Dense Networks, and verified in the Semi-physical simulation nodes.

Acknowledgement: This work was supported by Sichuan science and technology program (2019YFG0212) and China Postdoctoral Science Foundation (2019M653401).

References

Abdullahi, M.; Ngadi, M. A.; Dishing, S. I. (2017): Chaotic symbiotic organisms search for task scheduling optimization on cloud computing environment. *6th ICT International Student Project Conference*, pp. 157-162.

Breitgand, D.; Kutiel, G.; Raz, D. (2010): Cost-aware live migration of services in the cloud. *3rd Annual Haifa Experimental Systems Conference*, pp. 1-6.

Burge, J.; Ranganathan, P.; Wiener, J. (2007): Cost-aware scheduling for

heterogeneous enterprise machines. *IEEE International Conference on Cluster Computing*, pp. 481-487.

Cui, Q. M.; Gong, Z. Z.; Ni, W.; Hou, Y. Z; Chen, X. et al. (2019): Stochastic online learning for mobile edge computing: learning from changes. *IEEE Communications Magazine*, vol. 57, no. 3, pp. 63-69.

Garg, S. K.; Buyya, R.; Siegel, H. J. (2010): Time and cost trade-off management for scheduling parallel applications on utility grids. *Future Generation Computer*, vol. 26, no. 8, pp. 1344-1355.

He, Y. H.; Ren, J.; Yu, G.; Yu, G.; Cai, Y. (2019): D2D Communications meet mobile edge computing for enhanced computation capacity in cellular networks. *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1750-1763.

Hu, Y. C.; Patel, M.; Sabella. D. (2015): Mobile edge computing: a key technology towards 5G. *ETSI White Paper.*

IMT 2020 (5G) Promotion Group (2014): 5G network architecture design. *5G Network Architecture Design White Paper.*

IMT 2020 (5G) Promotion Group (2014): Typical 5G network capabilities. *5G Network Architecture Design White Paper.*

Ma, L.; Wen, X. M.; Wang, L. H.; Lu, Z. M.; Knopp, R. (2018): An SDN/NFV based framework for management and deployment of service based 5G core network. *China Communications*, vol. 15, no. 10, pp. 86-98.

Ning, Z. L.; Wang, X. J; Huang, J.; Yu, G. (2019): Mobile edge computing-enabled 5G vehicular networks: toward the integration of communication and computing. *IEEE Vehicular Technology Magazine*, vol. 14, no. 1, pp. 54-61.

Richart, M.; Baliosian, J.; Serrat, J.; Gorricho, J. L. (2016): Resource slicing in virtual wireless networks: a survey. *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462-476.

Rost, P.; Breitbach, M.; Roreger, H.; Erman, B.; Mannweiler, C. et al. (2018): Customized industrial networks: network slicing trial at hamburg seaport. *IEEE Wireless Communications*, vol. 25, no. 5, pp. 48-55.

Shen, S.; Deng, K.; Iosup, A. (2013): Scheduling jobs in the cloud using on-demand and reserved instances. *European Conference on Parallel Processing*, pp. 242-254.

Sun, Y. H.; Peng, M. G.; Mao, S. W.; Yan, S. (2019): Hierarchical radio resource allocation for network slicing in fog radio access networks. *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3866-3881.

Tiwari, A.; Nagaraju, A.; Mahrishi, M. (2013): An optimized scheduling algorithm for cloud broker using adaptive cost model. *3rd International Advance Computing Conference*, pp. 28-33.

Vo, P. L.; Nguyen, M. H.; Le, T. A.; Tran, N. H. (2018): Slicing the edge: resource allocation for RAN network slicing. *IEEE Wireless Communications Letters*, vol. 7, no. 6, pp. 970-973.

Wen, R.; Feng, G.; Tang, J. H.; Quek, T. S.; Wang, G. et al. (2019): On robustness of

network slicing for next-generation mobile networks. *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 430-444

Xiong, K.; Leng, S.; Hu, J.; Chen, X. S.; Yang, K. (2019): Smart network slicing for vehicular Fog-RANs. *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3075-3085.