5 th International Conference on Control & Signal Processing (CSP-2017)
Proceeding of Engineering and Technology –PET
Vol.25 pp.44-47

A Rapid Heuristic For The Virtual Machines Migration Scheduling Problem

Fatma Moalla
ISG tunis
University of Tunis,Tunisia
Email:fatma.moalla.fen@gmail.com

Ali Balma
National Higher Engineering School of Tunis
University of Tunis, Tunis, Tunisia
Email: alibalma05@yahoo.fr

Mehdi Mrad
Department of indistrual Engineering
College of Engeneering
King Saud University,Riyadh,Saudi Arabia
Email:mmrad@ksu.edu.sa

***Abstract*—This paper aims to develop a heuristic for the virtual machines migration problem in geographically distributed cloud data centers. The algorithm objective is to find a migration sequence of virtual machines with the minimum cost corresponding to the backbone traffic. Our heuristic consists of finding at each iteration the order of migration that minimizes the maximum traffic between the two considered data centers. Computational results show that the proposed heuristic is rapid and efficient.**

I. INTRODUCTION

Due to the technological progress over the last decade, cloud computing has appeared as a new paradigm providing access to multiple resources such as servers and applications. Virtualization technologies enable an effective resource management whithout loss of performance. Its basic advantage consist of dividing physical resources in a manner that allows a better workload assignment. So that, logical resources are able to run applications like physical ones. Virtual Machine (VM) is defined as a computer system simulation with consolidating capabilities.

Cloud computing services which are infrastructure as service (Iaas), software as service (SaaS), and platform as service (PaaS) became popular. This emphases the network application requirements. The amount of traffic for service communication affect the performance of the system. As an illustration case, we tackle the migration problem in a geographically distributed data center which involves a huge amount of traffic. In this context, an effective migration scheduling of VMs should be considered in the way that the main backbone traffic would be minimized [1]. This optimization can avoid network link congestion. Consequently, the problem gain more and more importance in the cloud community. The resulting migration traffic value should permit a trade-off between costs and performance.

In In geo-distributed cloud systems, the VMs connected to a data center are moved sequentially to the new one in order to minimize the customer services disruption. A bridge is installed serving as a gateway for the traffic between the migrated and the non-migrated VMs.

Let consider a backbone network with two data centers. Virtual machines are connected to data centers. We migrate virtual machines from the current data center to the other one by one (Fig.1). The following example consists of a network of 3 VMs with 3 data flow pairs (Fig.2). A valuated edge represents the traffic amount between the VMs. Let consider two different migration scenarios. The first consists of migrating first VM 2. So the incurred traffic will be 15. Then we move VM 3, the traffic would be 7. The last VM to migrate is 1. Consequently, the maximum value of the traffic throughout the migration process is 15 (Fig.3 and Fig.4).

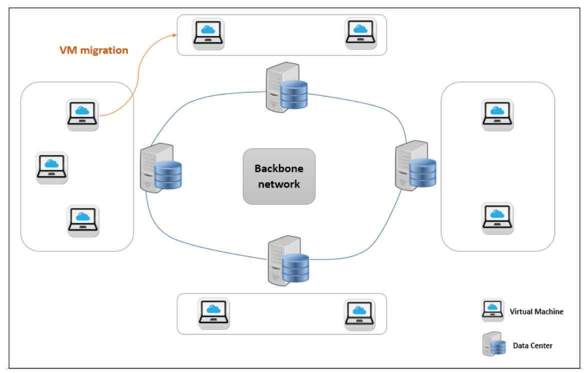

Fig. 1. VMs migration problem in geographically distributed DCs.

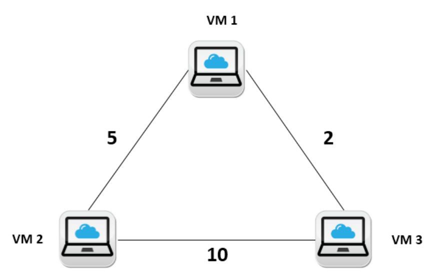

Fig. 2. Data traffic matrix for 3 VMs.

We propose a second migration scenario where we move first VM 3, the second migrating VM is 2. Finally VM 1 is migrated (see Fig.5 and Fig.6). The maximum traffic value for this scenario is 12 which involves less capacity than the first scenario.

This example indicate how significant could be the VMs order through the migration process .

ISSN 2356-5608

5 th International Conference on Control & Signal Processing (CSP-2017)
Proceeding of Engineering and Technology –PET
Vol.25 pp.44-47

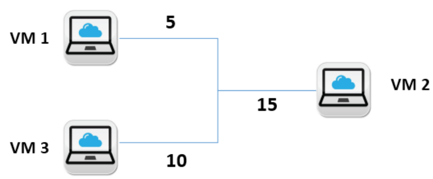

Fig. 3. first step of the first migration scenario.

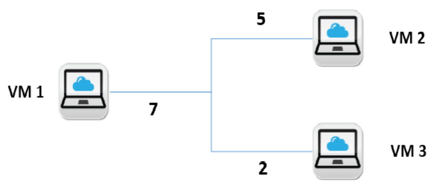

Fig. 4. second step of the first migration scenario.

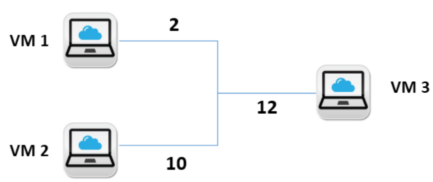

Fig. 5. first step of the second migration scenario.

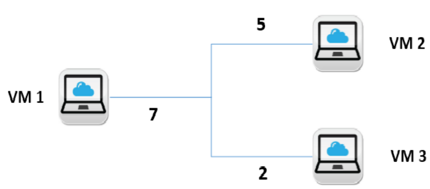

Fig. 6. second step of the second migration scenario.

This work proposes the following contributions:

- extend the work for the migration scheduling problem in [1] by proposing a rapid heuristic providing minimal cost for the entire backbone traffic.
- provide a comparaison context by considering this problem analogically to the migration scheduling problem in an access network [2] ;

The paper reports in section 2 a literature survey about VMs scheduling. Section 3 details the proposed method. Computational results are presented in section 4. Finally we conclude in section 5.

II. LITERATURE SURVEY

The VM migration problem is nowadays widely applied in the cloud computing field. Many works address it. A recapitulation of different works considering virtual machine migration is presented in [3]. Another survey [4] addresses the migration mechanisms for virtual machines which are essentially the process migration, memory migration, and suspend/resume migration. A more recent survey in [5] give a global view on this problem. It tackles the virtualization technologies and presents a classification of current VM migration schemes and compares between them. The authors in [6] Aim to reduce the network traffic by proposing an efficient scheme dealing with VM dependencies and the importance of the network topology. The migration planning problem formulated in [7] as Constraints Satisfaction Problem aims to generate a migration step order while respecting the security and dependency needs. A novel multiobjective VM migration algorithm is presented in [8]. This problem studies application dependencies in order to decrease the migration network traffic. The paper [9] emphasizes the importance of networking cost among all the costs.

III. HEURISTIC

In this section, the gradual migration of the VMs updates, at each iteration, the current traffic value and propose the most eligible migrating VM. An approximative maximum traffic value is computed and a near optimal migration sequence is presented. Based on this estimation, a minimum cost is concluded based on this VMs order.

VM migration problem in geo-distributed cloud systems has many resemblances with the called access node network migration, which aims at migrating nodes from an outdated network to a new one [2]. Analogically to the access network, two data centers already placed represent the former and new access networks. VMs are equivalent to the set of migrating nodes. Formally, we can consider a graph as G = (V,E) where V is the set of virtual machines and E the set of links connecting these VMs. d_{it} represent the data flow between VMs i$\in$ V and t$\in$V. Let S2 be the set of already migrated virtual machines. The remaining VMs belong to the set S1. Obviously, as an initial state, the subset S1 is full and S2 is empty. We move at each step The VM that needs the minimum of capacity while migrating from S1 to S2. Multiple steps are performed aiming to compute the maximum traffic flow value. The procedure of our heuristic algorithm is given in algorithm 1

steps 1-7 are iterated for moving nodes from S1 to S2. We repeat the move steps, respecting the rule (1), until it remains only one node to migrate at the end.

$$Q(i) = \sum_{t \in S1} D_{it} - \sum_{t \in S2} D_{it} \quad (1)$$

Vt and fmax denote the traffic value and its maximum. The vertex with the minimum Q(i) is added to S2. The quantity fmax is updated at each iteration and finally computed to cost using different boards. At the end, the final output will be

ISSN 2356-5608

Algorithm 1 heuristic procedure

Input: $S1$= { 1,2,..,N } $S2 = \emptyset$
Output: updated maximum traffic cost value ;
Begin
  Step 1: Vt:= 0 fmax= 0 ,
  Step 2: finding the machine that requires the smallest capacity to migrate from S1 To S2

$$Q(i)=\sum_{t \in S1} D_{it} - \sum_{t \in S2} D_{it}$$
$$\text{min= Min}(Q_i)\ \forall i \in S1$$
$$\text{j=Argmin}(Q_i)\ \forall i \in S1$$

  Step 3:move VM j
  Step 4: update: Vt = Vt+min;
  Step 5: if (Vt $>$ fmax) then fmax = Vt;
  Step 6: if ($|S1| > 1$) go to step 2 otherwise STOP.
  Step 7: compute the cost of fmax
End

the updated value of fmax. In step 1, we initialize the Vt and fmax quantities. In step 2, we look for finding the minimum value for Q(i) which refers to the difference in traffic amount resulted by the relation of the virtual machine i with the other subset machines and the relation of VM i with the other VMs in the same subset. The selected VM in step 2 will be moved to the other subset (step 3). Vt will be updated by adding the minimum quantity of traffic corresponding to the selected VM (step 4). We compare this updated value to the maximum traffic value fmax and we stop if we had only one node to move in the corresponding subset. We return at the end the cost of maximum traffic value using the different board types.

IV. COMPUTATIONAL EXPERIMENTS

Our heuristic is compared to the formulations in [2]. We select our benchmark for instances inferior and equal to 40 from [2] as well as the board instances.We choose randomly the instances of hundreds of VMs. The overall Instances sets consist of a traffic matrix randomly generated between 100 Mb/s and 1 Gb/s and considering only the symmetric traffic. The instances consider only the complete graph. Consequently, it will be always a traffic amount between each pair of VMs. The heuristic has been coded in c language and executed on a i7 processor machine with 1,8 GHz. Memory and 8 GB Ram. The running times unit is the second. Computational results for comparison between exact method and our heuristic are summarized in table I to IV corresponding respectively to $|V|$ =25, 30, 35 and 40. Results for relatively large size instances are presented in table V and VI. For each table, the column are detailed as follows:

- SEQ-INST: the sequence instance
- H: heuristic solution value
- T_H: running time
- OPT : exact solution value from [2]
- T_{OPT}: exact solution running time from[2]
- G: is the gap between the heuristic H and the optimal solution OPT computed in this way: $G = 100 * (H - OPT)/OPT$.

TABLE I
RESULTS FOR 25 VM

SEQ-INST	H	T_H	OPT	T_{OPT}	G
S-1	3.6	0.01	3.6	0.83	0
S-2	3.6	0.02	3.6	0.34	0
S-3	3.6	0.02	3.6	0.17	0
S-4	3.6	0	3.6	1.09	0
S-5	3.6	0.02	3.6	0.41	0
S-6	3.6	0	3.6	0.80	0
S-7	3.6	0.02	3.6	0.50	0
S-8	3.6	0	3.6	0.05	0
S-9	4.6	0.02	3.6	3.22	27.77
S-10	4.6	0	3.6	0.39	27.77

TABLE II
RESULTS FOR 30 VM

SEQ-INST	H	T_H	OPT	T_{OPT}	G
S-1	5.4	0	5.4	0.64	0
S-2	5.4	0	5.4	0.08	0
S-3	5.4	0	5.4	1.52	0
S-4	5.4	0	5.4	1.30	0
S-5	5.4	0	5.4	0.34	0
S-6	5.4	0	5.4	0.27	0
S-7	5.4	0	5.4	0.08	0
S-8	5.4	0	5.4	0.86	0
S-9	5.4	0	5.4	1.19	0
S-10	5.4	0	5.4	0.08	0

TABLE III
RESULTS FOR 35 VM

SEQ-INST	H	T_H	OPT	T_{OPT}	G
S-1	7.2	0	7.2	0.66	0
S-2	7.2	0.01	7.2	0.67	0
S-3	7.2	0	7.2	0.69	0
S-4	7.2	0	7.2	0.59	0
S-5	7.2	0	7.2	0.55	0
S-6	7.2	0	7.2	0.58	0
S-7	7.2	0	7.2	0.55	0
S-8	7.2	0	7.2	1.39	0
S-9	7.2	0	7.2	0.63	0
S-10	7.2	0	7.2	0.61	0

TABLE IV
RESULTS FOR 40 VM

SEQ-INST	H	T_H	OPT	T_{OPT}	G
S-1	9	0.01	9	6.91	0
S-2	9	0	9	15.44	0
S-3	9	0	9	10.11	0
S-4	9	0	9	6.36	0
S-5	10	0	10	>3600	0
S-6	10	0	10	>3600	0
S-7	10	0	9	44.44	11.11
S-8	10	0	10	>3600	0
S-9	10	0	9	16.13	11.11
S-10	10	0.02	10	2578.19	0

The heuristic proved its efficacity by proving optimality in major cases. The running time, considered with value 0 (under 0.00s) is insignificant. For instances of thousands of VMs, the heuristic is still performant contrarily to exact method which is not able to solve large instances as mentioned in in table V and VI.

ISSN 2356-5608

TABLE V
RESULTS FOR 100 VM

SEQ-INST	H	T
S-1	57.6	0
S-2	59.4	0.01
S-3	59.4	0
S-4	59.4	0.02
S-5	59.4	0.02
S-6	59.4	0
S-7	59.4	0.02
S-8	59.4	0
S-9	59.4	0.02
S-10	59.4	0

TABLE VI
RESULTS FOR 200 VM

SEQ-INST	H	T
S-1	237.6	0.03
S-2	238.6	0.05
S-3	238.6	0.03
S-4	238.6	0.05
S-5	238.6	0.03
S-6	238.6	0.03
S-7	238.6	0.05
S-8	238.6	0.03
S-9	238.6	0.05
S-10	238.6	0.03

V. CONCLUSION

The Virtual Machine Migration Scheduling Problem is addressed. Its objective is to find a migration sequence of migrating VMs. The heuristic consists on moving the VMs from a data center to another one while minimizing the maximum backbone traffic. This heuristic is considered simple and rapid. Our future work can deal with finding other effective heuristics for this problem.

REFERENCES

[1] H. Teyeb, N. Ben Hadj-Alouane, S.Tata and A. Balma *Optimal Dynamic Placement of Virtual Machines in Geographically Distributed Cloud Data Centers*. International Journal of Cooperative Information Systems, 2017.
[2] M. Mrad, A. Balma, F. Moalla, and T. Ladhari, *Nodes Migration Scheduling of Access Networks*. IEEE Trans. Network and Service Management14(1): pp 77-90,2017.
[3] C. T. Joseph, K. Chandrasekaran, R.Cyriac, *A perspective study of virtual machine migration*. Advances in Computing Communications and Informatics (ICACCI 2014 International Conference on, pp. 1499- 1503, 2014
[4] V. Medina and J.M. Garcia, *A survey of migration mechanisms of virtual machines.*. ACM Comput. Surv. 46, 3, pp..30-33, 2014
[5] R. W.Ahmad, A. Gani, S. Hafizah Ab. Hamid, M. Shiraz, F. Xia, S. A. Madani, *Virtual machine migration in cloud data centers: a review taxonomy and open research issues*. The Journal of Supercomputing, vol. 71, pp. 2473, 2015, ISSN 0920-8542.
[6] V. Shrivastava, P. Zerfos, K.-W. Lee, , H.Jamjoom, Y.-H.. Liu, and S. Banerjee, *Application aware virtual machine migration in data centers*. In Proceedings of INFOCOM, 2011 (pp. 6670). 2011.
[7] S. Al-Haj, E. Al-Shaer, *A formal approach for virtual machine migration planning*. Network and Service Management (CNSM) 2013 9th International Conference on, pp. 51-58, 2013.
[8] D. Huang, Y.Gao, F.Song, D.Yang, H. Zhang, *Multi-objective virtual machine migration in virtualized data center environments*. Communications (ICC) 2013 IEEE International Conference on, pp. 3699-3704, 2013.
[9] T. Yapicioglu and S. Oktug, *A traffic-aware virtual machine placement method for cloud data centers*. in Proc. 2013 IEEE/ACM 6th Int. Conf. Utility and Cloud Computing, ser. UCC 13. IEEE Computer Society, Washington, DC, USA (2013)

ISSN 2356-5608