Column Generation based-Approach for IaaS Aware Networked Edge Data-Centers

Abdallah Jarray[1], Javier Salazar[1,2], Ahmed Karmouch[1], Jocelyne Elias[2], Ahmed Mehaoua[2]
[1]SITE, University of Ottawa, 800 King Edward, Ottawa, Ontario, K1N 6N5 Canada
[2]LIPADE, Université Paris Descartes-Sorbonne Paris Cité, 45 rue des Saints Pères, Paris, France
Email: ajarray, jsalazar @uottawa.ca, karmouch@site.uottawa.ca, jocelyne.elias@parisdescartes.fr

***Abstract*—In this paper, our focus is on proposing a resource allocation approach for Infrastructure as Cloud Service (IaaS). Networked Edge Data-Centers (DCs) were selected as a physical infrastructure to handle IaaS Cloud requests. Doing so, we hold the potential of providing both a relatively high degree of independence between physical data centers' outages and an opportunity to reach economically IaaS cloud service users with low latency. An IaaS request is defined through two main requirements: Hosting and Networking resources. In literature, a number of works have proposed IaaS mapping approaches, however their focus was mainly on the cloud hosting requirements. Consequently, this may result in a high blocking of IaaS requests and a low utilization of resources inside the DCs due to stranding and fragmentation. To handle these drawbacks, we propose a joint optimization model of DCs and networking IaaS requirements that makes use of Column Generation technique coupled with a Branch and Bound approach in order to solve it efficiently.**

***Index Terms*—Cloud service, IaaS, edge data-center, cloud provider, resource allocation, optimization, Integer Linear Programming, column generation.**

I. Introduction

Network virtualization has been received recently significant attention as it allows to setup a cost-effective data centers infrastructure for storing large volumes of data and hosting large-scale service applications. Nowadays, large companies like Google, Facebook, and Amazon have been made large investment in massive virtualized data centers supporting cloud services that require large-scale computations and storage [1]. With the emergence of cloud computing models (CloudNaaS, EC2, SS3, etc.), service hosting in data centers has become a profitable business that plays a crucial role in the future of Internet [2]. Cloud model will allow to put forward new classes of applications according to the paradigm 'as a service' ,i.e., software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). However, despite of the adopted cloud service model, ultimately the goal of cloud computing providers is to create a fluid pool of virtual resources across networked cloud sites that enables the flexibility of infrastructure provisioning in terms of configuration, accessibility and availability for the user.

The networked Cloud architecture is built using virtualized data-Centers (VDCs), where the role of the traditional ISP is separated into: a Cloud Provider (CP) and Service Providers (SPs). The CP is the business entity that owns and manages the physical infrastructure of networked DCs. The CP leases virtualized Data-center resources to multiple SPs. Nowadays, an area of rapid innovation in the industry of cloud services is the deployment of edge data centers having on the order of thousands of servers [3]-[4]. Highly interactive or Office production applications are a natural fit for edge data centers placed in the last mile closer to major population centers. Doing so, propagation delay will be minimized and the dollar cost of communication (network transit cost) would go down since servers are located closer to the end-user. Moreover, these micro data centers can be used as nodes in content distribution networks and other distributed applications, such as email [5].

In literature, many approaches have implemented optimization techniques for resource allocation in cloud computing, while most proposals [6], [12] and [17] have focused on designing heuristic-based algorithms or on restricting the mapping problem to only addressing the problem of Virtual Machines (VM) allocation into physical machines. Fewer works [7], [8], [9] and [18] have focused on geographically distributed architecture where network features such as bandwidth or jitter play an important role in the requested IaaS service.

The aforementioned limitations motivate us to propose an approach called CG-IaaS for Column Generation based-approach to handle IaaS requests. CG-IaaS is a resource allocation approach for IaaS requests in the context of networked edge data centers. To reap economic benefits from geo-diversity, our approach CG-IaaS manages edge data centers and network resources as a joint optimization problem. Doing so, we hold the potential to provide both a relatively high degree of independence between physical data center outages such as power and an opportunity to reach cloud service users with low jitter and latency [10]. Indeed, the geographic diversity of edge data centers can be used as a source of redundancy to improve system availability, as not all geographically distributed sites are likely to have a power outages at the same time.

Moreover, since IaaS mapping problem is known to be NP-hard [14], our approach proposes a mathematical model that makes use of the Column Generation (CG) technique [20]. The proposed CG-IAAS model decomposes the IaaS mapping problem into a master problem which takes care of constraints related to substrate resources availability, and a pricing problem which includes constraints related to mapping of IaaS requests.

The remainder of the paper is organized as follows. Section II presents other related work to our proposal. Section III defines the IaaS Cloud mapping problem. Section IV presents the CG-IaaS approach. Section V introduces benchmarks and lists the proposed performance evaluation metrics, followed by the numerical results. Finally, Section VI concludes the paper.

II. Related Work

A number of approaches have been proposed in the literature to handle the challenging IaaS Cloud mapping problem. Most proposals have focused on either relaxing networking resources requirement or adopting heuristic two-phase mapping approach [6], i.e., data center resources and networking resources are considered in two separate optimization phases.

Papagiani *et. al* [8] investigate a two-phase mapping approach for hosting and networking IaaS resources, which may ensue in a high blocking of IaaS requests and a low utilization of resources inside the data centers due to stranding and fragmentation.

Authors in [7] proposed and an optimization algorithm based on a multi-objective formulation which optimizes the used power as well as the load balancing among DC servers. Nevertheless, cost of networking equipments is not considered in the modeling, which lacks the realistic evaluation of the economical benefit of user IaaS requests.

In [11] a Bin packing approach is proposed to optimize the dynamic allocation of Virtual Machines (VMs) into Physical Machines (PMs). However, it only considers the allocation of required computing CPU resources. Therefore, important IaaS resources as memory, storage and networking requirements are not taken into account in the optimization model. Furthermore, a geographically distributed cloud requires a modeling that includes the DC sites's locations and networking requirements, such as bandwidth, latency, etc.

In [12], authors use data collected from Google DCs and use different pricing model, in consequence, they have real prices for energy on different DCs. The proposed model is very efficient to optimize the CP profit's, however it lacks the network optimization component which is an important factor to fulfill the networking customer QoS requirements.

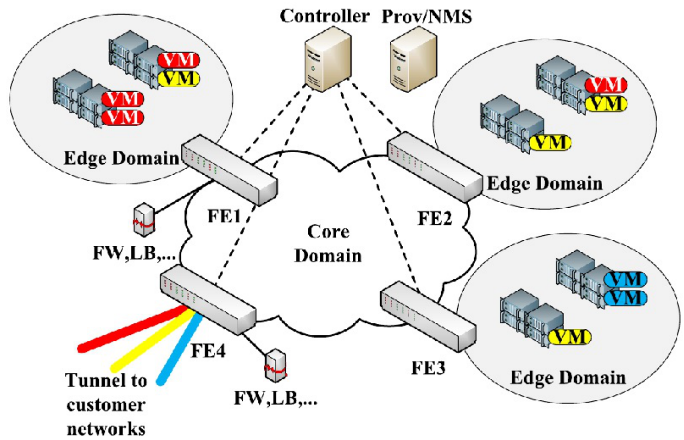


Figure 1: An example of Networked Edge DCs [15]

A joint optimization model has been proposed in [13] that groups IaaS, SaaS and Cloud users' requirements into one model in order to achieve a common welfare for the whole cloud participants. The results of this approach are quite interesting, however a multi-objective modeling requires more analysis of the Pareto model distribution optimum, i.e., analysis of the weighting that provides the optimal IaaS, SaaS and cloud users' objectives.

III. IaaS Cloud Mapping Problem Description

A. Networked Edge Data Centers Infrastructure

We distinguish two classes of DC-based Cloud architecture, namely, Networked Edge DCs (see Figure 1) and the large geographically distributed DCs. Large DCs enjoy economy-of-scale and high manageability due to their centralized nature, however they have their inherent limitations when it comes to service hosting. Economic factors impose that there will be only built in locations where construction and Operational Expenditures (OPEX) are low. Accordingly, large data centers are generally located far away from end-users, and this leads to a higher communication cost and non-respect of QoS requirements in terms of delay and throughput. To overcome these drawbacks Edge DCs have been put forward, for instance, EdgeCloud , Micro-DCs, NanoData, etc. This new class of small-scale DCs adapt well for service hosting at the network access/edge networks, where services can be hosted close to the end-users. Therefore, we selected Netwoked Edge DCs as the main repository for cloud resources that will be used to serve IaaS user requests.

B. Mathematical Modelling

As aforementioned, we adopt a Networked Edge DCs infrastructure to handle IaaS user requests. We represent the DC physical infrastructure by an undirected graph $G_d = (S_d, H_d, L_d)$, where S_d denotes the set of backbone switching nodes, H_d the set of DC server locations (hosting nodes), and L_d the set of network links. Each physical link between DC server locations $l \in L_s$ offers a bandwidth capacity b_l. Similarly, each Datacenter hosting node $u \in H_d$ offers a set of attributes: computing capacity p_u, memory capacity m_u, and a storage capacity s_u.

Similarly, a Cloud IaaS request is denoted by a Virtual Network I_n, where $n \in \mathcal{N} = \{1, 2, \ldots, N\}$ and represented by a directed graph $G_n = (A_n, S_n, E_n)$, where A_n denotes the set of virtual hosting nodes, S_n the set of virtual switching nodes and E_n the set of virtual networking links. Each virtual hosting node $a \in A_n$ has: computing requirement $p(a)$, memory requirement $m(a)$ and storage requirement $s(a)$. Also each virtual link $e \in E_n$ has a bandwidth requirement $b(e)$.

C. IaaS Cloud Mapping

The mapping of each IaaS request can be decomposed into hosting and network mapping as follows.

1) IaaS Hosting: Each IaaS virtual hosting node $a \in A_n$ from the same IaaS request n is mapped to different substrate hosting node $u \in H_d$ by mapping: $M_{\text{N}} : A_n \mapsto H_d$. Similarly, each switching node $s \in S_n$ from the same IaaS request n is mapped to different substrate switching node $v \in S_d$ by mapping $M_{\text{N}} : S_n \mapsto S_d$.

2) IaaS Inter-DC Sites Networking: Similarly, each virtual link $e \in E_n$ from the same IaaS request n is mapped to a set of substrate path $\pi^e_{uv} \subset \Pi^s$ by mapping $M_{\text{L}} : E_n \mapsto \Pi^s$, where (u, v) are substrate nodes assigned to virtual nodes (s, d) source and destination nodes of virtual link e, respectively.

D. CP Objective Function

When an IaaS request arrives, the CP has to determine whether to accept or reject. The main guideline of his decision will be based on both the availability of Networked Edge DCs resources and the economic benefit (cost) of accepting an IaaS request. As, in this paper, we focus on computing, storage, memory and bandwidth as the main substrate resources, we propose to calculate the mapping cost of each IaaS request n, $G_n = (A_n, S_n, E_n)$, as follows.

$$\text{COST}[I_n] = \text{COST}\left[M_{\text{N}}(A_n), M_{\text{N}}(S_n), M_{\text{L}}(E_n)\right] \quad (1)$$

E. IaaS Request Modelling

In the context of a IaaS cloud service, it seems reasonable that a small delay can be tolerated between IaaS request and setup. Accordingly, mapping of IaaS requests will be done by small-batch at each new planning period [19]. Hence, IaaS demand can be described as following, let T be the set of planning periods of time, indexed by $t \geq 1$ and $I(0)$ the initial set of IaaS requests, indexed by t. At the beginning of period $t \in T$, the set of IaaS requests is defined by:

$$I(t) = I(t-1) + I_{\text{NEW}}(t) - I_{\text{DROP}}(t) \quad (2)$$

where $I(t-1)$ is the set of accepted IaaS requests at the ending of period $t-1$. $I_{\text{NEW}}(t)$ is the set of new incoming and $I_{\text{DROP}}(t)$ is the set of ending IaaS requests at the outset of period t. Where NEW and DROP are randomly selected between 5% and 30%, giving us a range of cases from slowly fluctuating (5%) to fast changing (30%) of IaaS demands.

IV. Column Generation Formulation for IaaS Resource Allocation (CG-IaaS)

To overcome the complexity issue and calculate an optimal/near-optimal solution in acceptable computation time, we are proposing an approach called CG-IaaS that uses Column Generation technique [20]. CG-IaaS formulates the IaaS cloud mapping problem in terms of Independent Cloud Mapping Configurations (ICMCs), where each ICMC provides an IaaS mapping solution of a set of IaaS Requests. We denote by C the set of all possible ICMCs. Accordingly, the IaaS mapping problem can be formulated with respect to the variables $(\lambda_c), c \in C$, where variable λ_c denotes the if a configuration c is used or not. Thus, the networked cloud mapping problem under the new formulation consists to select a maximum of $|N|$ ICMCs, as in the best case, we serve all IaaS requests, where each IaaS request is granted by a distinguished ICMC. An ICMC configuration $c \in C$ is defined by the vector $(a^c_n)_{n \in N}$ such that: $a^c_n = 1$ if the ICMC c serves IaaS request I_n and 0 otherwise. We denote by COST_c the cost of configuration c. It corresponds to the costs of the used substrate resources (bandwidth, computing, memory, and storage) for mapping IaaS request granted by ICMC c. It is therefore defined as follows:

$$\text{COST}_c = \sum_{l \in L_c} b^c(l) \times c_b(l) + \sum_{u \in H_c} p^c(u) \times c_p(u) + m^c(u) \times c_m(u) + s^c(u) \times c_s(u)$$

where $b^c(l)$, $p^c(u)$, $m^c(u)$ and $s^c(u)$ are the used bandwidth, computing, memory and storage resources by ICMC c, respectively. We note that $c_b(l)$ is the unit bandwidth cost of link l and $c_p(u)$, $c_m(u)$ and $c_s(u)$ are, respectively, the unit cost of cpu, memory and storage in node u. Also, $L_c \subset L_d$ and $H_c \subset H_d$ define respectively the set of network links and hosting nodes used by ICMC c.

Using the Column Generation technique means that the IaaS cloud mapping problem is decomposed into a master problem which takes care of the constraints related to substrate resources availability, and a pricing problem which includes the constraints related to the mapping of IaaS resources. We present in the following master and pricing problem formulations.

A. Master Problem

The master problem corresponds to the choice of a maximum of $|N|$ configurations among the generated ICMCs, in order to minimize the objective function (Equation (3)). The proposed mathematical model is denoted by $ILP(M)$, is as follows.

1) Objective Function:

$$\min \sum_{c \in C} \text{COST}_c \, \lambda_c \tag{3}$$

2) Constraints:

$$\sum_{c \in C} -b^c(l) \times \lambda_c \geq -b_l; \qquad l \in L_d \ (\alpha_l) \tag{4}$$

$$\sum_{c \in C} -p^c(u) \times \lambda_c \geq -p_u; \qquad u \in H_d \ (\beta_u) \tag{5}$$

$$\sum_{c \in C} -m^c(u) \times \lambda_c \geq -m_u; \qquad u \in H_d \ (\gamma_u) \tag{6}$$

$$\sum_{c \in C} -s^c(u) \times \lambda_c \geq -s_u; \qquad u \in H_d \ (\eta_u) \tag{7}$$

$$\sum_{c \in C} \lambda_c \, a_c^n \geq 1; \qquad n \in N; \ (\psi_n) \tag{8}$$

$$\lambda_c \in \{0, 1\} \tag{9}$$

Equations (4), (5), (6) and (7) guarantee respectively the available physical bandwidth, computing (CPU), memory and storage resources. Equation (8) guarantees the satisfaction of IaaS requests with respect to available resources. Equation (9) expresses the integrality of master variable λ_c.

B. Pricing Problem

As mentioned previously, the pricing problem corresponds to the problem of generating an additional configuration (ICMC). It is defined as follows: Let α_l, β_u, γ_u, η_u and ψ_n be the dual variables associated with constraints (4), (5), (6), (7), and (8) respectively. Then, the reduced cost of variable $\lambda_c, c \in C$ can be written:

$$\begin{aligned}\overline{\text{COST}}_c = \text{COST}_c &+ \sum_{l \in L_d} \alpha_l \times b^c(l) \\ &+ \sum_{u \in H_d} \beta_u \times p^c(u) + \sum_{u \in H_d} \gamma_u \times m^c(u) \\ &+ \sum_{u \in H_d} \eta_u \times s^c(u) \\ &- \sum_{n \in N} a_c^n \times \psi_n. \end{aligned} \tag{10}$$

We now express (10) in terms of the variables of the pricing problem. Those variables are defined as follows. $z_n = 1$, if IaaS request I_n is served by ICMC c and 0 otherwise. $x_\pi^e = 1$, if virtual link $e \in E_n$ is assigned to path π and 0 otherwise. $x_u^a = 1$ if virtual hosting (resp. switching) node $a \in A_n$ (resp. $s \in S_n$) is assigned to physical node $u \in H_d$ (resp. $v \in S_d$) and 0 otherwise. We next derive the following relations between the above variables of the pricing problem and the coefficients of the master problem. For each $c \in C$ and $n \in N$, we have:

$$a_c^n = \frac{1}{|E_n|} \sum_{e \in E_n} \sum_{(u,u') \in H(s,d)} \sum_{\pi \in \pi_{uu'}^e} x_\pi^e$$

where $H(s,d) = H_d^2 \cup S_d^2 \cup H_d \times S_d$ is the set of all possible couples of source and destination that can be used for the mapping of any virtual link in the IaaS requests.

For each link $l \in L_d$, we have:

$$b^c(l) = \sum_{n \in N} \sum_{e \in E_n} \sum_{(u,u') \in H(s,d)} \sum_{\pi \in \pi_{uv}^e} b(e) \delta_\pi^l x_\pi^e$$

where $\delta_\pi^l = 1$ if path π uses link l and 0 if not.
For each node $u \in H_d$, we have:

$$p^c(u) = \sum_{n \in N} \sum_{a \in A_n} p(a) x_u^a$$

$$m^c(u) = \sum_{n \in N} \sum_{a \in A_n} m(a) x_u^a$$

$$s^c(u) = \sum_{n \in N} \sum_{a \in A_n} s(a) x_u^a$$

Accordingly, the reduced cost (10) can then be expressed by a linear expression.

Constraints:

a) Mapping of IaaS Hosting and Switching Nodes:

i. Mapping is done for all nodes of an accepted I_n .

$$z_n \leq \sum_{(u,u') \in H(s,d)} x_s^u \, x_d^{u'} \ ; \ (sd) = e \in E_n, n \in N.$$

ii. A virtual hosting node a of an IaaS I_n can be assigned to only one physical hosting node u.

$$\sum_{u \in H_d} x_u^a \leq z_n \ ; \ a \in A_n, \ n \in N.$$

iii. A virtual switching node s of an IaaS I_n can be assigned to only one physical switching node v.

$$\sum_{v \in S_d} x_v^s \leq z_n \ ; \ s \in S_n, \ n \in N.$$

b) Mapping of IaaS Networking Link:

$$x_s^u x_d^{u'} \leq \sum_{\pi \in \Pi^e_{uu'}} x^e_\pi \; ; \; (u,u') \in H(s,d) \; (sd) = e \in E_n.$$

At least one mapping path π is selected between a couple of substrate nodes (u, v) assigned to end virtual nodes (s, d) of a virtual link $e \in E_n$.

$$\sum_{(u,u') \in H(s,d)} \sum_{\pi \in \Pi^e_{uu'}} x^e_\pi \leq K \times z_n \; ; \; e \in E_n, \; n \in N.$$

For reliability purpose, a maximum of K mapping paths can be assigned to each virtual IaaS networking link of an accepted request I_n.

V. Numerical Results

A. Simulation Benchmarks

To evaluate the performance of CG-IaaS approach we use the two following benchmarks:

- Bin packing [11] (BIN-IaaS), where hosting and network requirements are mapped using a Bin per type of IaaS resource, i.e., computing Bin, bandwidth Bin, Storage Bin and memory Bin.
- Greedy hosting node mapping combined with a K-shortest path algorithm (G-IaaS) [6].

B. Experiment Setup

To evaluation the efficiency of the proposed periodical CG-IaaS model, we carried out experimental assessments using CPLEX [21]. We consider a physical infrastructure of four edge data centers connected through the NSFNet topology as a backbone network that includes 14 nodes located at different cities in the United States [18]. In each IaaS request, the number of virtual nodes is randomly determined by a uniform distribution between 2 and 20. The minimum connectivity degree is fixed to 2 links. QoS requirements of new IaaS requests are randomly determined by a uniform distribution among $J_1 = 5$ QoS classes for VN nodes and among $J_2 = 5$ QoS classes for VN links. Bandwidth/CPU/memory/storage unit cost, are expressed in terms of $\$X$, which represents the price of 1 Mb of bandwidth or 1 unit of CPU or 1 GB.

C. Performance Evaluation Metrics

To evaluate the performance of our CG-IaaS approach, we are measuring the following metrics.

c) IaaS demands' blocking ratio: measured as the ratio between the number of rejected IaaS requests and the number of the whole IaaS demands.

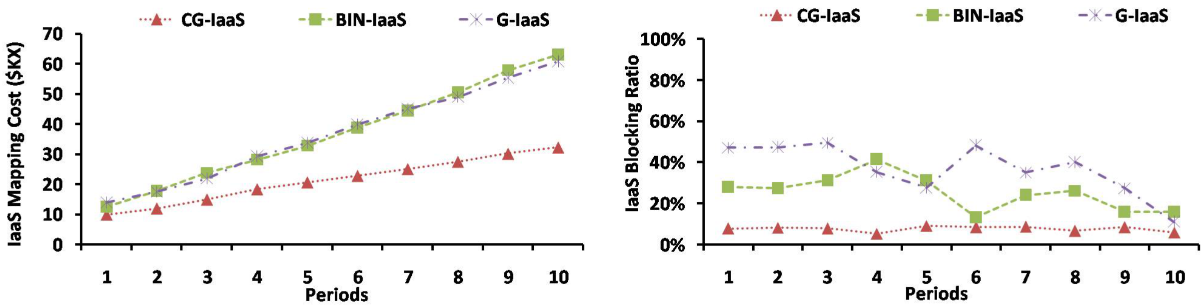
d) Bandwidth/CPU/Memory/Storage utilization: measured as the ratio between the used and the overall available bandwidth/CPU/Memory/Storage amounts.

D. Evaluation Results

Through this Section, we study the performance of the proposed CG-IaaS model compared to benchmarks in terms of IaaS blocking ratio, and bandwidth, CPU, memory and storage usage. Figure 2a plots the resulting cumulative CP IaaS mapping cost vs. the allocation time periods. In this Figure, we compare the IaaS mapping cost for CG-IaaS and the benchmark models BIN-IaaS and G-IaaS. The results show that CG-IaaS model provides the lowest mapping cost. The cost gap between our proposed embedding approach and benchmark varies from 5% to 35%. Figure 2b shows the blocking ratios of IaaS requests vs. the allocation time periods. CP accepts all IaaS requests, since their bandwidth, CPU, memory and storage requirements fit with the available amounts of these resources. This is one of the guidelines of the access control and reservation mechanism of substrate network resources in order to uphold QoS guarantees of accepted IaaS requests in previous period and still lasting in the current period. Figure 3a plots the percentage of bandwidth utilization vs. the allocation time periods. In this figure, we show that CG-IaaS model provides the highest bandwidth utilization. Indeed, the CG-IaaS model provides on average an utilization of 52% of the networks' bandwidth resources through all the planning period of time, where Bin packing and Greedy mapping used an average of 41% and 42%, respectively. The explanation of this tendency is straightforward as CG-IaaS model accepts the mapping of more IaaS requests as shown in Figure 2b. Figure 3b plots the percentage of substrate nodal CPU utilization vs. the allocation time periods. CG-IaaS model shows an average utilization of 40% of nodal CPU resources through all the planning period of time. The Bin Packing and Greedy mapping approaches use an average of 50% and 28% of available nodal CPU resources respectively. We observed the same results for memory and storage substrate resources, respectively, and we omit to show figures for these resources due to lack of space. In fact, results showed in terms of hosting resources usage confirm our expectation that the Greedy and Bin packing-based IaaS mapping approaches result in high blocking of IaaS requests, and a lack of profit due to bandwidth scarce. This is tightly related to the myopic hosting resources mapping that did not coordinate the requirements in terms of bandwidth, CPU, memory and storage.

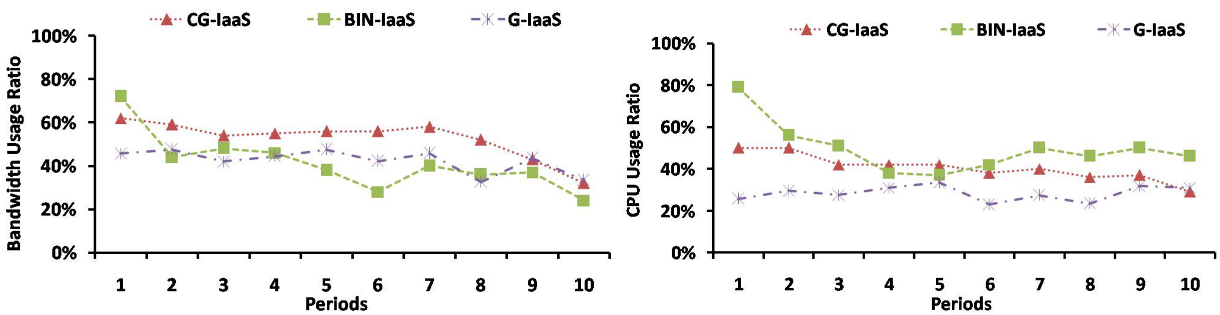
VI. Conclusion

To handle complexity issues of IaaS Cloud mapping problem, we proposed a Column Generation formulation, that decomposes the IaaS cloud service problem into a set of sub-problems easy to solve efficiently. The proposed approach performs a joint optimization of hosting and networking IaaS resources. Doing so,



(a) (b)

Figure 2: Mapping cost and IaaS requests blocking



(a) (b)

Figure 3: Periodical substrate resources usage

Cloud users satisfaction ratio and Cloud Provider's profit are increased substantially, since networked edge DCs resources are utilized efficiently.

REFERENCES

[1] E. Escalona, S. Peng, R. Nejabati, GEYSERS: A Novel Architecture for Virtualization and Co-Provisioning of Dynamic Optical Networks and IT Services, Future Network and Mobile Summit (FutureNetw), 2011.

[2] D.F. Bari, R. Boutaba, R. Esteves, L.Z. Granville, M. Podlesny, M.G. Rabbani, Q. Zhang, and M.F. Zhani, Data Center Network Virtualization: A Survey, IEEE Communications Surveys and Tutorials 15(2):909-928 (2013).

[3] Google docs and spreadsheets. URL http://docs.google.com.

[4] Microsoft office live. http://office.live.com.

[5] K. Church, J. Hamilton, and A. Greenberg, On delivering embarassingly distributed cloud services, Hotnets, 2008.

[6] Y. Zhu, M.H. Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components," IEEE Infocom, pp. 1-12, Apr. 2006.

[7] L., Z. Zeng, and X. Ye, Multi-objective Optimization Based Virtual Resource Allocation Strategy for Cloud Computing, IEEE/ACIS, 2012.

[8] C. Papagianni, A. Leivadeas, S. Papavassiliou, V. Maglaris, C. Cervell o-Pastor and A. Monje, On the optimal allocation of virtual resources in cloud computing networks, IEEE Transactions on Computers, Vol.62, No. 6, 2013, pp. 1060-1071.

[9] A. Tzanakaki, M. Anastasopoulos, K. Georgakilas, Energy Efficiency in integrated IT and Optical Network Infrastructures: The GEYSERS approach, Workshops Infocom, 2011.

[10] A. Greenberg, J. Hamilton, D.A. Maltz, P. Patel, The Cost of a Cloud: Research Problems in Data Center Networks, ACM SIGCOMM Computer Communication, Vol.39, Issue 1, pp. 68-73, January 2009.

[11] L. Eyraud-Dubois, H. Larcheveque, Optimizing Resource allocation while handling SLA violations in Cloud Computing platforms. IEEE 27th IPDPS, 2013.

[12] L. Rao, X. Liu, L. Xie and W. Liu, Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-Electricity-Market Environment, Infocom, 2010.

[13] C. Li and L. Li, Efficient resource allocation for optimizing objectives of cloud users, IaaS provider and SaaS provider in cloud environment, The Journal of Supercomputing, Vol. 65, Issue 2 , pp 866-885, 2013.

[14] D. Anderson, *Theoretical approaches to node assignment*, Unpublished Manuscript, http://www.cs.cmu.edu/ dga/papers/ andersen-assign.ps, 2002.

[15] F. Hao, T. V. Lakshman, S. Mukherjee, H. Song, Secure Cloud Computing with a Virtualized Network Infrastructure. HotCloud'10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, 2010.

[16] T. Wood, K.K.T. Ramakrishnan, P. Shenoy and J. Vander Merwe, Enterprise-Ready Virtual Cloud Pools: Vision, Opportunities and Challenges, USENIX HotCloud workshop, 2009.

[17] Q. Zhang, Q. Zhuz, M.F. Zhani, R. Boutaba and J.L. Hellersteinx, Dynamic Service Placement in Geographically Distributed Clouds, ICDCS, 2012.

[18] A. Amokrane, M.F. Zhani, R. Langar, R. Boutaba and G. Pujolle, Greenhead: Virtual Data Center Embedding across Distributed Infrastructures. IEEE Transactions On Cloud Computing, Vol. 1, No. 1, January-June 2013.

[19] A. Jarray *et al.*, DDP: A Dynamic Dimensioning and Partitioning model of Virtual Private Networks resources, Computer Communications: Vol. 35, pp: 906-915, 2012

[20] M.E. Ubbecke, and J. Desrosiers, *Selected Topics in Column Generation,*, Operations Research, vol. 53, pp. 1007-1023, 2005.

[21] IBM ILOG Cplex 12.4, User Manual, http://pic.dhe.ibm.com/ infocenter/ cosinfoc/v12r4/topic/ilog.odms.studio.help/pdf/ usrcplex.pdf, last visited July 2013.