QoS-based Cloud Resources Partitioning Aware Networked Edge Datacenters

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*—This paper focuses on the resource allocation problem in the context of Cloud Computing. More specifically, this work considers the problem of optimizing the mapping cost of Infrastructure as Cloud Service (IaaS) onto a Networked Edge Data-Centers (DCs) with respect to Quality of Service (QoS) requirements. This work proposes to dynamically partition the networked DCs resources over IaaS requests belonging to different QoS classes. In literature, a number of works have proposed IaaS mapping approaches; however their focus was mainly on the cloud hosting requirements and do not take into account the dynamics of IaaS QoS requirements. Consequently, they may not offer QoS guarantees for accepted IaaS requests which may result in a higher customer dissatisfaction ratio. The originality of our work is in the forethought and the investigation of these issues. To do so, a column generation based-formulation is proposed coupled with the Branch and Bound technique in order to solve it efficiently. Doing so, this allows the Cloud Provider to: *(i)* minimize IaaS mapping cost, and *(ii)* calculate the optimal and dynamic partitioning of DCs resources to uphold QoS guarantees for IaaS requests.**

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

I. Introduction

The continue growing uptake of cloud computing as a widely accepted computing paradigm calls for novel approaches for designing the network architecture that supports several basic resources like computing, storage and bandwidth. Cloud computing and its different models based on the paradigm "as a service"; software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS), will allow to put forward new classes of applications. Network virtualization has been identified as the mainstay of the current and future success of cloud computing networks, as it will allow Service Provider to setup a cost-effective data centers infrastructure for storing large volumes of data and hosting large-scale service applications. Indeed, network virtualization allows multiple heterogeneous virtual network (VN) architectures to coexist on a shared physical infrastructure. Also, since VNs are logically separated, implementing performance isolation and application Quality of Service (QoS) is facilitated [2]. Therefore, providing scalability, on demand resource allocation and an efficient usage of datacenter resources. Network virtualization will fulfill the requirements of future cloud networks where clients are expected to be able to specify QoS and processing requirements for hosted applications.

With the emergence of cloud application models, service hosting in data centers has become a profitable business that plays a crucial role in the future of Internet [2]. 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. This flood of various resources will enable the flexibility of infrastructure provisioning in terms of configuration and gives the illusion of infinite resources availability for user.

Networked Clouds can be built using a classical data center architecture based on the classical ISP business model that uses dedicated virtualized servers/ Data-centers to run applications [10]. However, relying on the conventional architecture results in poor server utilization and high operational cost. Accordingly, the alternative consists on using virtualized data-Centers (VDCs), where the role of the traditional ISP is separated into: a Cloud Provider (CP) and Service Providers (SPs) [19]. 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, Cloud Provider will be more able to honor contracted SLAs regarding user QoS requirements. Indeed, for example 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], [13] and [11] have focused on restricting the mapping problem to only addressing the problem of Virtual Machines (VM) allocation into physical machines. Fewer works [7], [8], [9] and [19] have focused on geographically distributed architecture where QoS requirements 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-QoS-IaaS for Column Generation based-approach to handle IaaS requests with QoS guarantee. This is a resource allocation approach for IaaS requests in the context of networked edge data centers with bandwidth and computing QoS guarantees. This proposal allow CP to dynamically partition the networked DCs resources over IaaS requests belonging to different QoS classes. To reap economic benefits from geo-diversity, the allocation approach manages computing edge data centers and networking resources as a joint optimization problem. Doing so, it holds 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 QoS guarantee, e.g., reserved computing and networking requirements. Therefore, CP guarantees low QoS jitter and latency [10].

Moreover, since IaaS mapping problem is known to be NP-hard [15], our approach proposes a mathematical model that makes use of the Column Generation (CG) technique [22]. The proposed CG-QoS-IaaS model decomposes the IaaS mapping problem into a master problem which takes care of constraints related to the optimal partitioning of available substrate resources among QoS classes, and a pricing problem which includes constraints related to mapping of IaaS requests respecting QoS requirements.

Another important contribution of CG-QoS-IaaS approach that has not been addressed in the literature is the time-reservation constraint and QoS guarantee. Most references [7], [12], [13], [11] that propose dynamic embedding approaches relax QoS constraints or consider independent online variant of IaaS resource embedding problem, where acceptance in one period does not guarantee the IaaS request admission in the next one. Accordingly, there is no substrate resource reservation with QoS guarantees for IaaS requests that lasts for multi-periods. In this work, it is assumed that once allocated, substrate resources are reserved for the service duration that typically lasts for more than one period of time.

The remainder of the paper is organized as follows. Section II defines the related work and re-iterates on the objectives of this paper. Section III presents the adopted networked edge datacenters infrastructure and defines IaaS Cloud mapping problem. Section IV introduces the CG-QoS-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. The challenges are mainly related to increased computational complexity when IaaS data center and networking IaaS resources requirements are considered as joint optimization problem. To cope with this issue, mapping solutions in literature have focused on either relaxing QoS requirement by focusing only on the computing requirements [12], [11], [13], [9], [7] or by adopting a two-phase approach [8], [18].

Two-phase-based approach consists on preselecting in a first stage the mapping of IaaS hosting nodes. Network mapping (virtual links) is done in a second stage. Accordingly, the main drawbacks of this sequential approach are as follows:

1) Hosting node mapping is performed without considering its relation to networking mapping. Hence, non-join hosting and network mapping may result into high blocking of IaaS requests and under-utilization of Cloud DCs resources resulting in reduced profit for CP.
2) Mapping of IaaS resource requirements is done based on heuristic approaches, which may effect the optimality of the mapping solution.

Lionel *et al.* proposed a Bin packing approach that optimized dynamically the mapping of Virtual Machines (VMs) into Physical Machines (PMs). Therefore, networking requirements are not taken into account in the optimization model which may result in violation of QoS requirements.

Authors in [11] proposed a joint-VM provisioning approach in which multiple VMs are consolidated and provisioned together, based on an estimate of their aggregate capacity needs. This approach exploits statistical multiplexing among the workload patterns of multiple VMs. Unused cloud resources of a low utilized VM can be borrowed by other co-located VMs with high utilization, as the peaks and valleys in one workload pattern do not necessarily coincide with the others.

In [13], authors proposed an efficient model to optimize the CP profit's, however it lacks the network optimization component which is an important factor to fulfill the networking IaaS QoS requirements.

In [9] authors proposed a joint optimization model for hosting and networking resources. Using GEYSERS

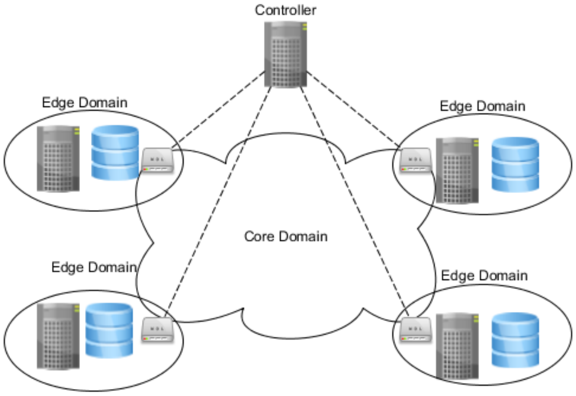


Figure 1: An example of Networked Edge DCs

[1] architecture and energy models they create a Mixed Integer Programming model that minimizes the energy consumption and cost. In this case the model is applied to optical networks and IT resources geographically distributed.

Authors in [7] proposed 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 and possible violation of QoS requirements.

Our main contribution will be on the forethought of the aforementioned limitations and meeting the following objectives.

- Take into account the dynamicity of IaaS request and their QoS requirements.
- For each accepted IaaS request, calculate the optimal one-shot networking and hosting scheme with respect to QoS requirements (latency, bandwidth, computing, and mapping locations).
- Use Cloud resources efficiently and minimize blocking rate of IaaS requests.
- Grant a time-reservation for allocated resource to accepted IaaS requests in order to uphold QoS guarantees.

III. QoS-based IaaS Cloud Mapping Problem Description

A. Networked Edge Data Centers Infrastructure

Data centers (DCs) have become an efficient and promising infrastructure for supporting the growth of data volumes and variety of Internet applications. These diversified network services and applications, e.g, cloud computing, data storage, P2P applications, require huge and critical resource demands from the physical substrate in terms of storage capacity, computing power, bandwidth, etc. Existing data center architectures lack the flexibility to effectively support these applications, which results in poor support of deployment and QoS requirements. Virtualized Data center (VDC) is considered as a promising solution to address these shortcoming. Indeed, virtualized data centers are designed in such a way to provide better management flexibility, better resources utilization, energy saving, and lower cost. Moreover, an important advantage of VDC is that physical resources are managed by a single infrastructure provider, which allows PIP to have a full view of the system resulting in an efficient resource allocation.

Two classes of DC-based Cloud architecture can be identified namely: *(i)* large geographically distributed DCs, and *(ii)* Networked Edge DCs (see Figure 1). Large DCs enjoy economy-of-scale and high manageability due to their centralized nature. However, Geographic distributed DCs have their inherent limitations when it comes to service hosting. Indeed, economic factors impose that there will be only built in locations where Capital and Operational Expenditures are low. Accordingly, large DCs are generally located far away from end-users which may result into non-respect of QoS requirements (e.g., End-To-end delay and throughput) as well as a higher networking cost. To overcome these drawbacks Edge DCs have been put forward (e.g., Micro-DCs and EdgeCloud). Another interesting example is the IBM private cloud that began in 1998 with the Nagano Olympics Website and became in US-Open 2013 an edge datacenter cloud that provides continuously available 3-sited services including Big Data and social media support. This new class of small-scale DCs adapts well for service hosting at the network access 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 requests.

B. Problem Statement

Infrastructure (IaaS) resources allocation is perhaps one of the most important aspects of cloud management since they are directly related to the costs and the QoS requirements. An efficient resource allocation would produce favorable impact in terms of profitability for the cloud provider. The resource allocation problem for cloud computing consists in minimizing infrastructure and communication costs while preserving QoS constraints. The QoS requirements are: *(i)* networking constraints where each IaaS link that belonging to a QoS class is defined by the required bandwidth and the end-to-end delay, and *(ii)* Hosting QoS constraints where each virtual node belonging to a QoS class is defined by the required computing capacity and the potential mapping locations.

IV. Column Generation-based Allocation Approach (QoS-CG-IaaS)

A. 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. Each Data-center hosting node $u \in H_d$ offers a computing capacity p_u.

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. The QoS requirements of each virtual link $e \in E_n$ belonging to class $j \in J_B$ are defined by the couple (b^j, d^j). b^j is the required bandwidth for QoS class j and d^j is the end-to-end delay of a routing path measured through the number of switching nodes between end points of the routing path. Similarly, QoS requirements of each virtual node $a \in A_n$ belonging to class $j \in J_U$ are defined by the couple (p_j, t_j). p_j is the required CPU for QoS class j and t_j is the set of potential embedding hosting locations that guarantees QoS requirements of class j.

B. 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 $M_{\mathrm{N}} : A_n \mapsto H_d$. 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 $M_{\mathrm{N}} : S_n \mapsto S_d$.

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

C. CP Objective Function

When an IaaS request arrives, the CP has to determine whether to accept or reject it. The main guideline of his decision will be based on QoS IaaS requirements, availability of cloud DCs resources and the economic cost of accepting an IaaS request. As, in this paper, we focus on computing 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.

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

D. IaaS Request Modelling

Mapping of IaaS requests will be done by small-batch at each new planning period [21], as it seems reasonable that a small delay can be tolerated between IaaS request and setup. Accordingly, the network planning time is divided into set of periods, representing a new IaaS allocation every δ units of time. The value of δ depends on the Cloud Provider objective in term of the balance between minimization mapping cost over time and the minimization of waiting time of IaaS requests to be setup. To uphold QoS requirements, each accepted IaaS request has the guarantee that allocated cloud resources are reserved for the duration of the full service that can stand for multi-periods of time. In this context, IaaS demand can be described with a set of requests, one for each new planning period of time. From one period of time to the next, it is assumed that a significant fraction of the IaaS requests remains the same, representing as an example the global steady state of the long term SLAs between the CP and its customers. The variation of the IaaS demand corresponds to the add or the drop of some IaaS requests. Each ending IaaS releases an amount of cloud Datacenter resources which can be reused to accept some new IaaSs. In more accurate manner, let P be the set of network planning periods of time, indexed by $p \geq 1$ and $R(0)$ the initial set of IaaS requests, indexed by r. At the beginning of period p, the set of IaaS requests is defined by:

$$R(p) = R(p-1) + R_{\textsc{add}}(p) - R_{\textsc{drop}}(p)$$

where $R(p{-}1)$ is the set of accepted IaaS requests at the beginning of period p, $R_{\textsc{add}}(p)$ (resp. $R_{\textsc{drop}}(p)$) is the set of new incoming (resp. ending) IaaS requests at the outset of period p. Where ADD and DROP are randomly selected between 5% and 30%, giving us a range of cases from slowly fluctuating dynamic traffic instances (5%) to fast changing dynamic traffic instances (30%).

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

Whenever we use a column generation technique, the original problem is decomposed into two subproblems: *(i)* master problem where the coefficients are implicitly defined and of which we always only solve a restricted form, i.e., with a restricted number of columns (ICMCs), and *(ii)* pricing problem which corresponds to the problem of generating an additional column to the constraint matrix of the restricted master problem, i.e., generates an ICMC that improves the current value of the objective function. An ICMC configuration $c \in C$ is defined by the vector $(a_n^c)_{n \in N}$ such that:

$$a_n^c = \begin{cases} 1 & \text{if an ICMC } c \text{ serves IaaS Request } I_n. \\ 0 & \text{Otherwise.} \end{cases}$$

We denote by COST_c the cost of configuration c. It corresponds to the costs of the used substrate resources (bandwidth and computing) for the mapping of IaaS Request granted by ICMC c. It is defined as follows:

$$\text{COST}_c = \sum_{l \in L_c} b^c(l) \times c_l + \sum_{u \in H_c} p^c(u) \times c_u$$

where $b^c(l)$ and $p^c(u)$ are the used substrate bandwidth and computing resources by ICMC c respectively. Also, L_c and H_c define the used set of physical links and hosting nodes respectively. Recall that using a column generation formulation means that the original problem is decomposed into a master problem and a pricing problem which corresponds. To do so, we define the following decision variables.

$$\lambda_c = \begin{cases} 1 & \text{if an ICMC } c \text{ is used in IaaS mapping.} \\ 0 & \text{Otherwise.} \end{cases}$$

b_l^j defines the amount of bandwidth to be setup on link l for QoS class $j \in J_B$.
p_u^j defines the amount of computing capacity to be setup on hosting node u for QoS class $j \in J_U$.

1) 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 (2)). The proposed mathematical model is denoted by $MIP(M)$, is as follows.

a) Objective Function:

$$\min \sum_{c \in C} \text{COST}_c \, \lambda_c + \sum_{l \in L_d} \sum_{j \in J_B} c_l \times b_l^j + \sum_{u \in H_d} \sum_{j \in J_U} c_u \times p_u^j \tag{2}$$

b) Constraints:

$$\sum_{c \in C} \lambda_c \, b_j^c(l) \le b_l^j; \qquad l \in L_d, j \in J_B \ (\alpha_l j) \tag{3}$$

$$\sum_{c \in C} \lambda_c \, p_j^c(u) \le p_u^j; \qquad u \in H_d, j \in J_U \ (\beta_u j) \tag{4}$$

$$\sum_{c \in C} \lambda_c \, a_c^n \ge 1; \qquad n \in N \qquad (\psi_n) \tag{5}$$

$$\lambda_c \in \{0, 1\} \qquad c \in C \tag{6}$$

$$b_l^j \in \mathbb{N}; b_l^j \le b_{max}; \qquad l \in L_d, j \in J_B \tag{7}$$

$$p_u^j \in \mathbb{N}; p_u^j \le p_{max}; \qquad u \in H_d, j \in J_U \tag{8}$$

Equation (3) defines the amount of bandwidth to be setup on a link l for the usage of IaaS requests belonging to QoS Class j. Variable Variable b_l^j is bounded by b_Max in order to guarantee some load balancing of traffic among network links. Equation (4) defines the amount of computing capacity to be setup on a node u for the usage of IaaS requests belonging to QoS Class j. Variable p_u^j is bounded by p_Max in order to guarantee some load balancing of traffic among network nodes. Equation (5) guarantees the satisfaction of IaaS requests with respect to available resources.

2) Pricing problem: A column generation formulation can be solved using a technique corresponding to an iterative solution process where one starts from a constraint sub-matrix (a set of columns), where each column is associated with a ICMC c and solve the associated so-called restricted master problem. An analysis of the solution of the restricted master problem throughout the so-called reduced costs is then conduced in order to check whether there exists a column with a negative reduced cost, i.e., a column whose addition could improve the value of the current restricted master problem. If this is the case, the resulting enlarged restricted master problem is solved again and the column generation technique iterates until the Linear Programming (LP) optimality condition is met, i.e., no more column with a negative reduced cost can be identified. The optimal LP solution only provides a lower bound on the optimal integer solution. We first formulate the pricing problem, where each solution of the pricing problem with a negative reduced cost corresponds to an improving ICMC.

As mentioned previously, the pricing problem corresponds to the problem of generating an additional configuration (ICMC), i.e., an additional column for the constraint matrix of the current master problem. It is defined as follows. Let $\alpha_l j$, $\beta_u j$ and ψ_n be the dual variables associated with constraints (3), (4) and (5) respectively. Then, the reduced cost of variable λ_c can be written:

$$\begin{aligned} \overline{\text{COST}}_c = \text{COST}_c + \sum_{l \in L_d} \sum_{j \in J_B} \alpha_l j \times b_j^c(l) \\ + \sum_{u \in H_d} \sum_{j \in J_U} \beta_u j \times p_j^c(u) - \sum_{n \in N} a_c^n \times \psi_n \end{aligned} \tag{9}$$

We now express (9) 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. $x_a^u = 1$ if virtual hosting node $a \in A_n$ is assigned to physical node $u \in H_d$ 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:

$$a_c^n = \sum_{e \in E_n} \sum_{(u,u') \in H_d^2 \cup H_d \times S_d \cup S_d^2} \sum_{\pi \in \pi_{uu'}^e} x_\pi^e$$

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

$$b_j^c(l) = \sum_{n \in N} \sum_{e \in E_n^j} \sum_{(u,u') \in H_d^2} \sum_{\pi \in \pi_{uv}^{ej}} b^j \delta_\pi^l x_\pi^e$$

For each node $u \in H_d$, we have:

$$p_j^c(u) = \sum_{n \in N} \sum_{a \in A_n^j} P_j x_a^u$$

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:

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

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 $e \in E_n$.

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

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

3) Solving the CG-QoS-IaaS Mathematical Model: Recall that the main objective of the CG model is to calculate the optimal QoS-based partitioning of networked edge data centers resource among QoS IaaS demand classes. $LP(M)$ denotes the continuous relaxations of the master problems $MIP(M)$, obtained by exchanging the integrality constraints (6) by $\lambda_c \in \mathbb{R}^+$ for any $c \in C$. Since the number of ICMC configurations is important then, $LP(M)$ is initialized by a subset of possible artificial configurations. Then, the restricted master problem is solved until optimality. To check the optimality of the obtained solution within the original problem, it is required to check the existence of a variable λ_c with a negative reduced cost. If such a variable exists then, it is added to the master problem and, it is solved again. Otherwise, $LP(M)$ is solved to optimality. To solve the initial $MIP(M)$ the following algorithm is used.

1) Relax the integrality of Data Center design variables as follows: $b_l^j \in \mathbb{R}, \ l \in L_d, j \in J_B$ and $p_u^j \in \mathbb{R}, \ u \in H_d, j \in J_U$.
2) Call procedure **Column_Generation()** to solve the resulting $LP(M)$ to optimality,
3) Convert the design variables back to integer format ($b_l^j \in \mathbb{N}$ and $p_u^j \in \mathbb{N}$) while keeping the variables λ_c continuous. The obtained mixed ILP program is denoted by $MIP(M)$.
4) Use the MILP solver of (CPLEX) to solve the resulting $MIP(M)$ program.
5) To calculate an integer solution, re-establish integrality constraint on variable λ_c and proceed with a branch-and-bound procedure using CPLEX package on selected columns in $MIP(M)$ solution.

Procedure Column_Generation()

1) Solve the $LP(M)$ master problem using CPLEX algorithm and go to Step 2.
2) Solve the pricing problem and go to Step 3.
3) Add the resulting column to the current master problem, and re-iterate with Steps 1 and 2 until no column can be found with a negative reduced cost. In such a case the master problem is solved to optimality.

V. Numerical Results

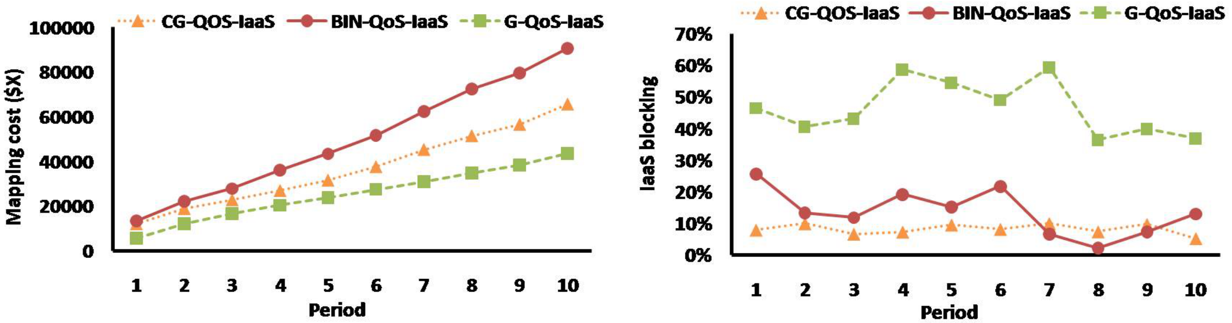
A. Simulation Benchmarks

To evaluate the performance of CG-QoS-IaaS approach the two following benchmarks are used:

- Bin packing [12] (BIN-QoS-IaaS), where Computing and Bandwidth requirements are mapped using a CPU Bin and bandwidth Bin respectively.
- Greedy computing node mapping combined with a K-shortest path algorithm (G-QoS-IaaS) [6].

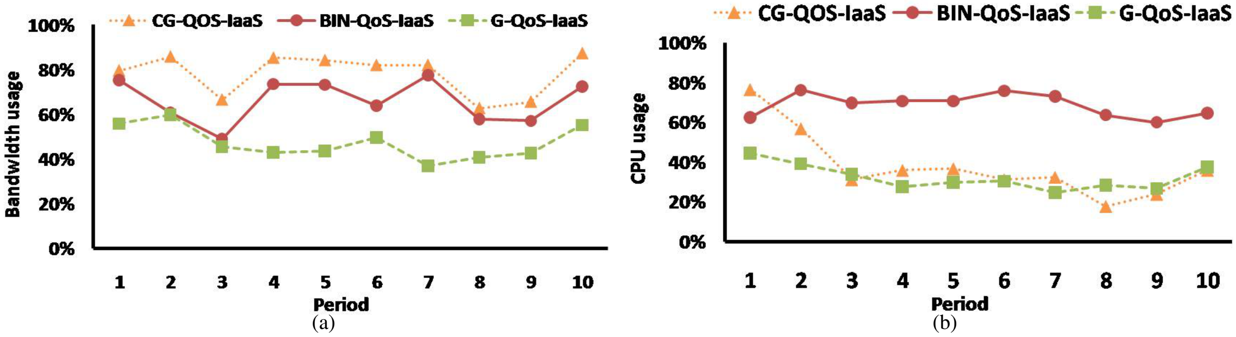
B. Experiment Setup

To evaluate the efficiency of the proposed QoS-based resource allocation model, experimental assessments are carried out using CPLEX MIP Solver [23]. A physical infrastructure of four edge data centers connected through the NSFNet topology is used [20]. The backbone network includes 14 nodes located at different cities in the United States [20]. In each IaaS request, the number of virtual nodes is randomly determined by a uniform distribution between 2 and 20. The minimum



(a) (b)

Figure 2: Mapping cost and IaaS requests blocking



(a) (b)

Figure 3: Periodical substrate resources usage

connectivity degree is fixed to 2 links. QoS requirements of new IaaS requests are randomly determined by a uniform distribution among $|J_B| = 5$ QoS classes for IaaS nodes and among $|J_U| = 5$ QoS classes for IaaS links. Bandwidth and computing unit costs are expressed in terms of $\$X$, which represents the price of 1 Mb of bandwidth or 1 unit of computing capacity.

C. Performance Evaluation Metrics

To evaluate the performance of CG-QoS-IaaS approach, following metrics are measured.

1) Mapping Cost measured as the cost of used substrate resources.
2) IaaS demands' blocking ratio measured as the ratio between the number of rejected IaaS requests and the overall number IaaS demand.
3) Bandwidth utilization measured as the ratio between the used and the overall bandwidth amounts.
4) Computing capacity utilization measured as the ratio between the used and the overall computing amounts.

D. Evaluation Results

Through this Section, we study the performance of the proposed CG-QoS-IaaS model compared to benchmarks in terms of IaaS blocking ratio, bandwidth and computing capacity 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-QoS-IaaS and the benchmark models BIN-QoS-IaaS and G-QoS-IaaS. The results show that G-QoS-IaaS model provides the lowest mapping cost. The cost gap between our proposed embedding approach and benchmarks varies from -35% to 35%.

Figure 2b plots the blocking ratio vs the allocation time periods. The results show that the greedy approach rejects between 36% and 59% of the requests, this includes the ones using QoS classes that require larger amounts of resources and therefore are more expensive. On the other hand the Bin-IaaS approach accepts more requests but still present an elevated blocking ratio on some periods and also larger cumulative cost. In this aspect the proposed CG model maintains a uniform blocking ratio through all time periods while reducing effectively the mapping cost guaranteeing the satisfaction of the QoS requirements.

Figure 3a plots the percentage of bandwidth utilization vs. the allocation time periods. In this figure, we show that CG-QoS-IaaS model provides the highest bandwidth utilization. Indeed, the CG-QoS-IaaS model provides on average an utilization of 78% of the networks' bandwidth resources through all the planning period of time, where Bin packing and Greedy mapping used an average of 66% and 47%, respectively. The explanation of this tendency is straightforward as CG-QoS-IaaS model maintains a higher utilization to cope with the entire different QoS requirements, the lower values presented on the other algorithms are consistent with their lower acceptance ratio.

Figure 3b plots the percentage of substrate nodal CPU utilization vs. the allocation time periods. CG-QoS-IaaS model shows an average utilization of 38% of nodal CPU resources through all the planning period of time. The Bin Packing and Greedy mapping approaches use an average of 68% and 32% of available nodal CPU resources respectively. In fact, results shown 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 scarcity. This is tightly related to the myopic hosting resources mapping that did not coordinate the requirements in terms of bandwidth and CPU usage.

VI. Conclusion

Using Operational Research Tools, i.e., ILP and Column Generation technique, a dynamic partitioning of networked DCs resources over IaaS QoS classes is proposed. The substrate network resources are allocated periodically through a MIP algorithm in order to accept profitable new IaaS requests, without disruption of the ones accepted in previous periods. The advantages of the proposed model lies in its ability of: *(i)* finding the optimal tradeoff point of minimizing the mapping cost and maximizing IaaS acceptance ratio, *(ii)* performing a joint optimization of hosting and networking IaaS resources while guaranteeing no-disruption of mapped demand in previous periods, and *(iii)* calculating the optimal partitioning of Cloud Edge DCs resources among QoS classes. Experiments were conducted using CPLEX Concert Technology environment. It was shown that CG-QoS-IaaS model outperforms benchmark approaches. On average, IaaS acceptance is increased up to 35%. Blocking of IaaS requests due to Edge DCs resources scarcity is reduced.

VII. Acknowledgment

This work is supported by a grant from Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] E. Escalona, S. Peng, R. Nejabati, GEYSERS: A Novel Architecture for Virtualization and Co-Provisioning of Dynamic Optical Networks and IT Services, 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] X. Meng *et al.*, Efficient Resource Provisioning in Compute Clouds via VM Multiplexing, Proceeding ICAC '10, Pages 11-20, ACM New York, NY, USA, 2010

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

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

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

[19] 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.

[20] S. Peng, R. Nejabati, and D. Simeonidou, Role of Optical Network Virtualization in Cloud Computing, IEEE/OSA Journal of Optical Communications and Networking. Vol. 5, No. 10, 2013, pp. 162-170.

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

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

[23] 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 September 2014.