Leveraging Network Virtualization for Energy-Efficient Cloud: Future Directions

Fatoumata B. Kasse and Bamba Gueye
Université Cheikh Anta Diop de Dakar
Dakar, Senegal

Halima Elbiaze
Université du Québec à Montréal
Montreal, Canada

***Abstract*—Reducing the energy consumed in cloud computing is becoming one of the most challenging research directions due to the overwhelming growth of services that are hosted and delivered by cloud computing. Indeed, the energy consumed by data transport represents a significant percentage according to the overall consumption of the cloud. Hence, by exploiting network and router virtualization technologies, we firstly propose a Green Cloud Architecture (GCA), where we can either shut down, or make in sleeping mode virtual routers; or migrate virtual routers towards another physical router according to energy-awareness. Secondly, we evaluate our green cloud architecture by proposing an energy-aware resource allocation algorithm. The mapping algorithm is evaluated through simulations and our green architecture significantly reduces the power consumption during data transport by up to 41%.**

I. Introduction

Network virtualization [1] can serve as a mainstay of the future Internet that allows service providers (SPs) to offer their resources as services to the general public. In contrast to grid technologies, hardware and platforms can be virtualized in cloud computing. Likewise, each user has unique access to its individual virtualized environment.

Despite clouds offer clear opportunities for enterprises to significantly reduce their growing data center and infrastructure technology hardware expenditures, the power consumed during transport and switching can be a significant percentage of total energy consumption in cloud computing [2]. In addition, the tendency is that network energy consumption follows Moore's law by doubling every 18 months and the current trend is to build data centers in geographic area with access to cheap power, or have cold temperature like arctic regions, and thus, the geographic distance between users and cloud will be lengthened. In such case, the energy consumed during the transport will be increased.

In this paper, we focus on power saving strategies during transport and switching in cloud computing. Our work makes the following contributions:

- **Green Cloud Architecture (GCA):** We propose a network architecture where router virtualization is the mainstay to transport the data. Therefore, physical routers can create multiple virtual routers, as well we design mechanisms to acquire and control network resources. Since the power consumed by a router depends on the number of activate ports [2], by enabling line cards or virtual routers instances to be dormant, we can reduce the power consumed by routers as well as the overall electricity consumption due to communications.
- **Energy-aware resource allocation algorithm design:** The Virtual Network Embedding (VNE) heuristic maps the virtual routers on top physical routers and seeks to minimize the power consumed during transport. In order to reduce the energy consumption during the communications, our energy-aware algorithm takes into consideration: *(i)* the heterogeneity issues between nodes and links in terms of power consumption and bandwidth capacity; *(ii)* the nodes and links assignment are done simultaneously; *(iii)* an admission control for checking whether the Virtual Network *(VN)* arrival could be satisfied or not; *(iv)* the possibility that Virtual Links *(VL)* can be split up over multiple physical paths in favour of reducing energy consumption as well as maximizing the traffic sent through the links, while minimizing the usage of total resources of the substrate network.

The remainder of the paper is organized as follows. In Section II, we survey the different studies related to energy-efficiency. Next, Section III presents a background of the designed metrics and illustrates our green network architecture that enables power savings during transport. Section IV evaluates the proposed GCA-VNE algorithm into heterogeneous substrate network. Finally, Section V concludes the paper and outlines our future work.

II. Related work

Several resource allocation and discovery approaches in network virtualization environment have been surveyed in [1]. Studies in [1] showed that when the virtual nodes *(Vnodes)* and virtual links *(VL)* are embedded jointly this allows a better mapping with lower cost and less elapsed time compared to the two stage embedding approach [3]. Su *et al.* [3] devised an embedding called "*EA-VNE*" which is based on *CPU* and bandwidth constraints Nevertheless, their algorithm does not support path splitting. Similarly, a resource allocation that takes into account the heterogeneity of the virtual and substrate networks was proposed in [4]. Their eliminative constraints for VNs embedding are based on the location, the load and number of CPU, the free *RAM* amount, etc.

The suggested embedding in [5] enables path splitting and link migration. Nevertheless, for a better embedding as suggested by [4], their approach [5] should correlate their link migration with the previous node-mapping. Basically, most of these algorithms fail to take into consideration the fact that the resources owned by both virtual and substrate network can be heterogeneous.

III. Leveraging routers virtualization for an energy-efficient transport

A. Background on design metrics

Purely, energy proportionality means that the energy consumed by a networking device should be proportional either to the load, or the number of active ports, or powered on line cards. In this sense, our proposed metrics take into consideration those factors. The first metric is related to the power consumed by each router that belongs to the substrate network. Note that, the terms "router" and "switch" are used interchangeably.

$$Power_{switch} = \sum_{r=0}^{configs} numports_{configs_r} \times Power_{configs_r} + num_{linecards} \times Power_{linecards} + Power_{chassis} \quad (1)$$

In fact, the power consumed by a router increases linearly according to the number of line cards ($num_{linecards}$) available into the router [6]. Since each line card can have several ports ($numports_{configs_r}$) running at the rate "*configs*", $Power_{configs_r}$ gives the power consumed by a port running at the rate r, and $numports_{configs_r}$ is the number of ports at rate r, where r can be 10 Mbps, 100 Mbps, or 1 Gbps. Port utilization is not considered in Eq. 1 since the traffic through networking devices does not have a significant effect [6].

The second metric, called *Energy Proportionality Index* (EPI), exhibits the potential correlation between the power consumed by a router and its load [2]. The EPI is equal to $\frac{M-I}{M} \times 100$, where M expresses the maximum power consumed, and I denotes the power consumed by the switch when it is turned on but does not process traffic (idle state). Basically, I returns the chassis power ($Power_{chassis}$ Eq. 1) consumption. In fact, the maximum power is the power consumed when all the components that are installed in the router are turning on.

Finally, the third metric called *Normalized power* enables to estimate the per-bit energy consumption during the transmission and switching. It is defined as $NormalizedPower = \frac{M}{G}$, where G represents the total aggregate bandwidth that it can support. Indeed, more the value of $NormalizedPower$ is less, more the router is energy-efficient. The EPI and the Normalized power metrics are used in order to estimate the energy proportionality of networking devices. These factors represent eliminative constraints throughout the selection of routers (substrate nodes).

B. Resource Allocation

A VNE deals with an efficient mapping of VNs onto physical network resources. The algorithm should find a set of Vnodes $\mathbb{N}^V$ onto a physical set of nodes $\mathbb{N}^S$, and a set of virtual links $\mathbb{E}^V$ onto a set of physical links $\mathbb{E}^S$. In other words, the substrate network is a graph $\mathbb{G}^S = (\mathbb{N}^S, \mathbb{E}^S)$, and the given VN to be embedded is a graph $\mathbb{G}^V = (\mathbb{N}^V, \mathbb{E}^V)$. Therefore, each substrate node (physical router) $n_i^S \in \mathbb{N}^S$ has an associated EPI value and a Normalized Power capacity. A substrate link $s = (n_i^S, n_j^S) \in \mathbb{E}^S$ between substrate nodes n_i^S, $n_j^S \in \mathbb{N}^S$ has an aggregate bandwidth capacity $BW(s)$.

a) ***Substrate path selection****:* Since the network resources of the substrate network are finite, each new satisfied VN reduces the residual substrate network resources. Consequently, the residual capacity of a link $i \in \mathbb{E}^S$ is defined as follows:

$$R_E(t^-, i) = \mathcal{BW}_{thres}(i) - \sum_{j=1}^{N_{vnet}} \sum_{l=1}^{N_{vlink_j}} ((LS_j^V(l_j) | l_j \supseteq i)) \quad (2)$$

where t^- is the time instance immediately before a VN arrival; N_{vnet} means the number of existing VN, whereas N_{vlink_j} is the number of VLs hosted by the VN j; LS_j^V, which represents the VL stress, returns the bandwidth allocated to the VL l_j.

If unconsumed bandwidth remains that can fulfill the actual VN needs, it should be used by the VL $l_j \in \mathbb{E}^V$. As long as it adequate unconsumed resources remains on a link $i \in \mathbb{E}^S$ (having the highest stress) and $i = 1 \ldots N_{Slink}$, where N_{Slink} is the number of links of the substrate network, one should start by assigning a given VL on top of this link. This embeding is performed only if we do not exceed the threshold bandwidth $\mathcal{BW}_{thres}(s) = \alpha \times BW(s)$ according to the required quality of service. The values of α are distributed between 0 and 1. Our goal is to avoid network congestion, and hence, respect delay constraints required by network applications.

The $\wedge$ function defined in Eq. 3 and used in Eq. 4, determines whether or not the link $i \in \mathbb{E}^S$ is an admissible path at time t.

$$\wedge(t, l_j, i) = \begin{cases} 1 & if \ R_E(t^-, i) \geq BW(l_j) \\ 0 & otherwise \end{cases} \quad (3)$$

where t^- is the time instance immediately before a VN arrival, and $BW(l_j)$ is the bandwidth requested by a given VL numbered l_j belonging to the j'th VN topology request arrival. Afterwards, the link stress ($\mathcal{LS}$) of a link $i \in \mathbb{E}^S$, which represents the link utilization rates, at time t is given by:

$$\mathcal{LS}(t, i) = \sum_{j=1}^{N_{vnet}} \sum_{l=1}^{N_{vlink_j}} ((LS_j^V(l_j) | l_j \supseteq i)) \wedge (t, l_j, i) \quad (4)$$

In contrast to [4], our $\mathcal{LS}$ defined in Eq. 4 takes into account the bandwidth previously allocated to each substrate link and the residual capacity bandwidth over this link. In fact, we leverage the residual bandwidth in order to weight each candidate link $i \in \mathbb{E}^S$.

b) ***Energy-aware resource allocation algorithm****:* In a similar way, we want to know if a given Vnode n_j (e.g., virtual router), belonging to the j'th VN, is active or not and

Algorithm 1 Energy-aware embedding algorithm (Upon i'th VN arrival)

Inputs:
$\mathbb{G}^S = (\mathbb{N}^S, \mathbb{E}^S)$: substrate topology;
$\mathbb{G}_i^V = (\mathbb{N}_i^V, \mathbb{E}_i^V)$: VNet topology;
Output: VNet$_{Embed}$ (Embedded Virtual Network)

1 Rank Vnodes $v \in \mathbb{N}_i^V$ according to their number of candidates $n \in \mathbb{N}^S$
2 Firstly assign nodes $v \in \mathbb{N}_i^V$ that have fewer substrate candidates nodes
3 **foreach** Node $v \in \mathbb{N}_i^V$ **do**
4   **foreach** Link $k \in \mathbb{E}_i^V$ connected to v **do**
5   LinkedVNode=GetLinkDestination(k)
6   **foreach** $SourceCandidate$ s in $v.Candidates$ **do**
7   CostNRG(s,d) = 0
8   **foreach** $DestCandidate$ d in *LinkedVNode.Candidates* **do**
9   PathCost(s,d) = CostNRG(s, d)
10   **end for**
11   CostNRG(s,d) = CostNRG(s,d) + $\frac{\sum_{e\in L(s,d)} \mathcal{LS}(t_i^-,e)\times\mathcal{NS}(t_i^-,s)}{Count(v.Candidates)}$
12   **end for**
13   **end for**
14   v.Embed = s | Path-cost is minimized with respect to power savings
15 **end for**

it is running on top the i'th physical node. In addition, the $\wedge$ function defined in Eq. 5, as well as used in Eq. 6, takes in consideration these requirements at a time t.

$$\wedge(t, l_j, i) = \begin{cases} 1 & if\ (n_j \supseteq i) \wedge (n_j\ is\ active) \\ 0 & otherwise \end{cases} \quad (5)$$

According to our node stress ($\mathcal{NS}$) formula (Eq. 6), we consider the energy proportionality of each physical router (EPI and $NPower$ metrics), and its instantaneous energy consumption ($Power_{switch}$). The $\mathcal{NS}$ of a given node $i \in \mathbb{N}^S$ gives its overall energy consumption. Following that, the $\mathcal{NS}$ of the i'th node at time t is:

$$\mathcal{NS}(t, i) = \frac{\sum_{j=1}^{N_{vnet}} \sum_{n=1}^{N_{vnode_j}} \wedge(t, n_j, i)}{\frac{1}{EPI} \times NPower} \times Power_{switch} \quad (6)$$

Indeed, the numerator of the first part of Eq. 6 returns the total number of Vnode instances running on the node $i \in \mathbb{N}^S$, where N_{vnode_j} represents the number of Vnodes that form the j'th VN. After the computation of node and link stresses, a ranking of all nodes following their stress is performed. Basically, the node $i \in \mathbb{N}^S$ having the maximum stress will be the first potential node amongst the set of candidates for hosting a Vnode. The path cost, of each potential path between the source candidate $s \in v.Candidates$ towards all other destinations $d \in LinkedVNode.Candidates$, is estimated by taking into consideration lines from 3 to 12 (Algorithm 1). Thereby, the candidate node that allows a path with energy savings (i.e lowest CostNRG) is selected (line 14).

IV. Performance evaluation

A. Simulation settings

Table I shows the average power consumed by the different components of two given routers. The line card consumptions illustrated in Table I are derived from [6]. According to Table I, the units of column labeled "*Power*" (resp. "*NPower*") is Watts (resp. Watts/Mbps). The term "*Qty*" returns the number of each line card type installed in the router. Also, "*FE*" and "*GE*" means Fast Ethernet and Gigabit Ethernet respectively. For the GSR 12008 (resp. Cisco 7507), its idle state consumes approximately 430 Watts (resp. 210 Watts) [6].

(a) Cisco GSR 12008

Line card type	Power	Qty	EPI (in %)	NPower
4 Port GE	92	2	43.12	0.058
4 Port OC-12/POS	72	1		
1 Port OC-48/POS	70	1		

(b) Cisco 7507

Line card type	Power	Qty	EPI (in %)	NPower
1 Port FE	26	3	42.77	0.085
4 Port GE	30	1		
1 Port 1.544 Mbps DS1	49	1		

TABLE I
Router power consumption summary.

We implemented our power-aware resource allocation algorithm in Matlab. Our discrete event simulator considers the same parameters depicted in [5] during the extensive simulation experiments. In this respect, the substrate network is a 100-node 354-link random topology generated by the *GT-ITM* tool. Physical nodes are chosen randomly as a GSR or a 7507 router. We assume that VN topology requests arrive in a Poisson process with an average rate $\lambda = 5$ VN requests per time unit. These requests are gathered during a fixed time window equal to 10 time units and processed at next time window as in [5]. In this setup, when the resource constraints of the Vns are satisfied they have an exponential service time with an average of $\mu = 10$ time units. Otherwise, the failed requests for instantiation of VNs will be enqueued and rescheduled at an appropriate time.

Besides, the number of Vnodes is uniformly distributed from 2 to 10, and each pair of Vnodes is randomly connected with probability 0.5 (*i.e.*, for n-node VN, we have $n(n-1)/4$ links). We set up an admission control mechanism. We run our simulation for 500 time units, which corresponds to about 2500 requests on average in one instance of simulation. The Vnodes can request a capacity bandwidth equal either to 100 Mbps or 1Gbps.

B. Results

Fig. 1 shows the node stresses of the overall node's network as well as the number of VNs mapped onto the substrate network over time. The left vertical axis gives the computed node stress for each node whereas the right vertical axis gives the number of embedded VNs. The horizontal axis illustrates the different time window over time. The shift noted with respect to both curves is due to the churn. Indeed, churn means the dynamic arrival and departure of VNs over time. In fact, when the VNs leave the network the used resources are released. Therefore, the node stresses of the physical nodes that hosted these VNs are reduced.

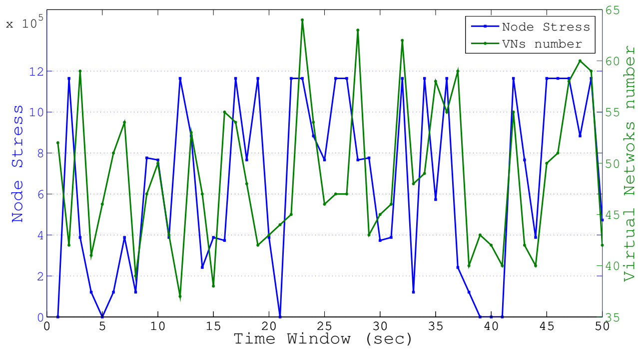
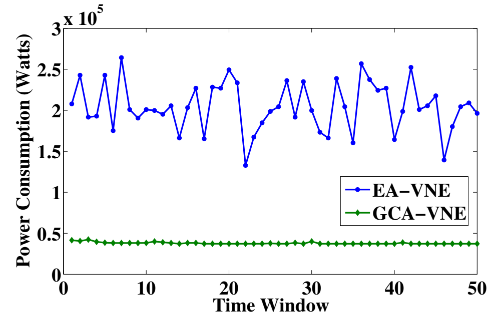


Fig. 1. Number of mapped virtual networks and Median Node Stress

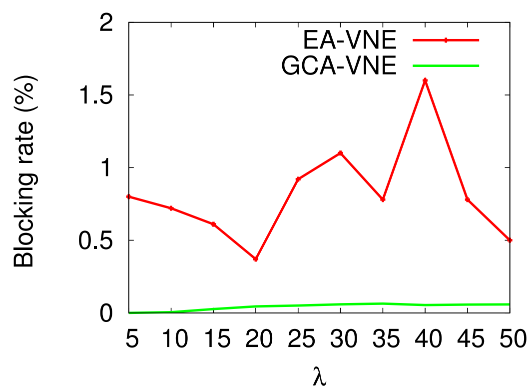
The blue curve labelled "Node Stress" (Fig. 1) represents the median percentile of node stresses overall node's network during each time window. The median values of node stresses exhibit a correlation according to the VNs assignment (green curve labelled "VNs number"). In this sense, the noted peaks with respect to the node stresses correspond to the instant where we note the maximum number of embedded VNs. Indeed, our algorithm instantiates firstly VR on physical nodes that have the highest stress. In addition, Fig. 1 depicts that for several time windows 50% of routers are in idle state (Node stress = 0) according to their power consumption.

In order to evaluate the performance of our algorithm, we implemented the *EA-VNE* algorithm [3]. We considered the same evaluation settings as used in [3], except the number of substrate nodes which is fixed at 100. Fig. 2(a) depicts the energy consumed by the overall network when the VNs are embedded according to an energy concern. A non-efficient network means a network where all networking devices are running according to their maximum energy. It is easy to verify that after 10 time windows the power consumption of GCA-VNE is in its steady state. The average energy consumption for GCA-VNE (resp. EA-VNE) is roughly equal to $37,000$ watts (resp. $196,000$ watts) whereas the energy consumed by a non-efficient network is always equal on average to $63,000$ watts (resp. $300,000$ watts). Indeed, GCA-VNE (resp. EA-VNE) can save up to 41% (resp. 35%) power cost with respect to a non-efficient network. This gain is due to the fact that GCA-VNE leverages path splitting.

Fig. 2(b) illustrates the percentage of rejected VN according to the average rate λ that varies from 5 to 50. Note that, the VN requests are gathered during a fixed time window 10, and processed at next time window. The percentage of rejected VN in EA-VNE is higher than GCA-VNE. When $\lambda = 5$ all arrival VNs are accepted for GCA-VNE approach in contrast to EA-VNE where 0.8% of virtual networks are rejected. Note that for λ values up to 40 the blocking rate is roughly in its steady phase with respect to GCA-VNE.



(a) Overall network power consumption



(b) Blocked Virtual Networks

Fig. 2. Comparison between GCA-VNE and EA-VNE.

V. Conclusion

This paper proposes a power-aware VNE algorithm that considers path splitting and the support of heterogeneous virtual and physical networks. Furthermore, GCA-VNE promotes the assignment of links and nodes simultaneously. The obtained results illustrate that our resource allocation algorithm (CGA-VNE) can save up to 41% of energy cost compared to an unaware-energy network. As future work we plan to take into consideration node and link migration, as well to deal with potential substrate link failures. Also, we will investigate the geographic location constraint that may happen during the resource allocation process.

Acknowledgment

Work done when the second author was with Université du Québec à Montréal as visiting scientist.

References

[1] A. Belbekkouche, M. Hasan, and A. Karmouch, "Resource discovery and allocation in network virtualization," *IEEE Communications Surveys & Tutorials*, no. 99, pp. 1–15, June 2012.

[2] A. P. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, "A survey of green networking research," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 1, pp. 3–20, 2012.

[3] S. Su, Z. Zhang, X. Cheng, Y. Wang, Y. Luo, and J. Wang, "Energy-aware virtual network embedding through consolidation," in *INFOCOM Workshops*, 2012, pp. 127–132.

[4] J. Nogueira, M. Melo, J. Carapinha, and S. Sargento, "Virtual network mapping into heterogeneous substrate networks," *IEEE Symposium on Computers and Communication*, pp. 438–444, June 2011.

[5] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Mar. 2008.

[6] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *Proc. INFOCOM*, April 2008, pp. 457–465.