International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)

http://ijcjournal.org/

An Adaptive Framework for Bandwidth Allocation in Cloud Networks

D Venkatesh[a], Puli Sukumar Reddy[b*]

[a]*Associate Professor in CSE, Gates Institute of Technology, GOOTY, A.P, India*
[b]*Research Scholar, Dept. of CSE, Gates Institute of Technology, GOOTY, A.P, India*
[a]*Email: deancseit@gmail.com*
[b]*Email: psukumarreddy@gmail.com*

Abstract

The increasing user demands for services in Virtual Machines and Cloud computing has made the resource allocation an impossible task to be manually performed by human operators. Software Defined Networks (SDNs) and Virtual Networks have been increasingly getting attention as a way for managing these configurations due to the abstraction of the controlling part of the network and its decoupling from the forwarding layer. In this partial paper, we summarize the paradigm of SDNs and Virtual Networks and survey the state of-the-art of dynamic resource allocation in the field of Virtual Machines and Cloud computing using SDNs and Virtual Networks.

Keywords: Cloud Networks; Framework; Bandwidth Allocation.

1. Introduction

The ongoing growth of user demands of Virtual Machines (VMs) and Cloud computing services is constantly increasing the complexity of managing these networks [6, 3, 1]. On the other hand, manually configuring the resources in these networks with variably demands is an impossible task. Therefore, Software Defined and Virtual Networks have been highlighted as the main solutions for reducing these demanding activities. SDNs decouple the controlling tasks of network operators from the underlying network infrastructure. Virtual Networks rely on abstracting the high-level policies from the low-level forwarding tasks. Using SDNs and Virtual Networks for dynamically configuring the resource usage is key for the success of the services and the Quality of Experience (QoE) of the users.

---

* Corresponding author.

In this partial paper, we survey the state-of-the-art of the dynamic resource allocation in SDNs and Virtual Networks in the field of Virtual Machines and Cloud computing. We provide an overview of Software Defined and Virtual Networks. After, we show the main characteristics of dynamic resource allocation and provide a survey of the main contributions in this area. Finally, we conclude this research and present our next steps for the full paper.

2. Software Defined and Virtual Networks

Software Defined Networks (SDN) have been increasingly getting attention of the computing research community over the past few years [6, 3]. Even though the concept is considered a new trend, its definition is already present since mid1990's, yet called Programmable Networks. The main goal of programmable networks has been to accommodate the growing needs of computing. This new paradigm has been created due to the always increasing number of protocols, standards and technologies that caused a growing difficulty of managing these different configurations. SDNs underley on decoupling the forwarding part of the network, i.e., the lower layers, from the management part of the network. In this approach, the management tasks are simplified with the abstraction of the lower layers. In order to make this possible, the SDN architecture consists on the data plane and the control plane. The data plane is responsible for the packet forwarding within the network while the control plane is responsible for the general decisions over the network. Network operators centrally perform the management tasks on the control plane while the network configures the forwarding tasks in the data plane according to these configurations.

The first challenge faced by SDNs has been enabling the programming of the data plane and integration to the control plane. This has been possible mainly by OpenFlow, the main SDN architecture. OpenFlow has enabled the separation between the controlling logic and the forwarding hardware. In a nutshell, OpenFlow enables control by providing access to the data plane of a network device in a one-size-fits-all way. In this approach, the OpenFlow controller is responsible for managing the flow table that contains the actions for each kind of message transferred in the network. When changes are needed, either administrators or applications can program directly in the control plane and OpenFlow reflects these changes in the data plane. The control plane can be viewed as a network operating system on which applications and administrators can program the network functionality.

Programming a network is an essential concept for Software Defined Networks. Virtual Networks also rely in this kind of high-level programmability in order to completely remove all interactions directly to the infrastructure, but mainly by abstractions [1].

When OpenFlow has enabled the data and control planes integration, the applications With in Software Defined and Virtual Networks became a field do explore. The first goal was mainly creating fully software defined LANs in a wide environment. Later on, improving the devices on the data plane, especially wireless, has increased the usage of SDNs and its programmability. Access points are abstracted in the data plane and mobility can be increased due to the independence of devices with the higher layers. On the other hand, the growing usage of Cloud Computing and Virtual Machines has pointed another challenge to SDNs. Enabling the

configuration and dynamic resource allocation in this kind of networks can be an outstanding advance on the way resources are used, especially with the continuously growth of cloud computing and its demands [7].

2.1 Dynamic Resource Allocation

Dynamic resource allocation consists in automating the allocation of resources in the data plane to users' applications by abstracting the low-level components involved, especially the underlying physical network, while guaranteeing fair allocation of such resources, transparency and scalability to the end users. In the context of SDNs and Virtual Networks, the main objectives of Dynamic Resource Allocation are overall cost reduction, for instance, by improving resource usage, and management overhead reduction for service providers [7]. Efficient techniques for resource allocation should address metrics such as Quality of Service (QoS) towards resource utilization, cost and power consumption reduction [7]. This section presents some of the dynamic resource allocation techniques [2,4,5,8].

2.1.1 Joint Allocation and Scheduling of Network Resource for Multiple Control Applications in SDN

Reference [2] Proposed a price-based joint allocation model for SDNs considering link bandwidth and flow table, given their limited capacities, as key network resources. The authors used the price paid by the user for both bandwidth and flow table capacities as a way of guaranteeing fair allocation of these key network resources. Bandwidth allocation was achieved by maximizing the sum of each control application in a logarithmic rate, where the control application rate is proportional to the bandwidth price for that particular application. Flow table allocation favors flows with higher hit-rate to optimize OpenFlow switches' throughput. On the other hand, it is also introduced a weight in the selection of such flows in order to prevent starvation of small flows. Based on virtual memory management, they designed a module to implement the allocation and scheduling of network resources for multiple applications in SDNs called Virtual Forwarding Space (VFS). VFS is implemented in the control plane and is responsible for allocating Virtual Forwarding Pages (VFPs), whose function is caching control instructions for each control application, as shown on Figure 1.

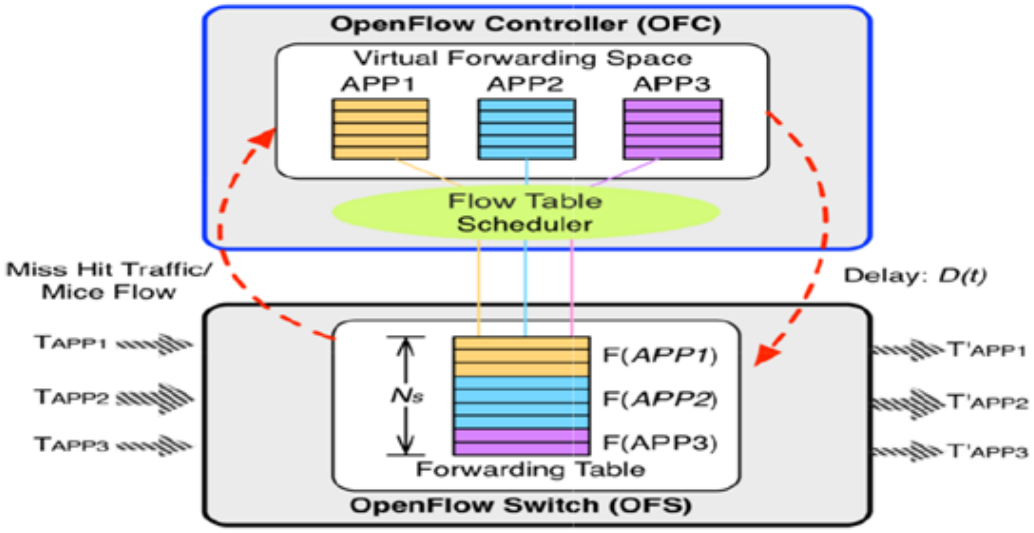

Figure 1: Flow table allocation for multiple applications in SDN

The proposed model has been evaluated against datasets provided by Chinese organizations and compared with a random scheduling model. The results have shown that the model reduces global delay time, i.e. packet forwarding time, by using the VFS decisions based on the hit-rate of a flow, as shown on Figure 2. In the comparison of their model with the random scheduling model, they have proven that the delay time of their solution is lower than the random scheduling. On the other hand, this comparison has lacked reliability since both models have been developed within the paper and no comparison between other proposed solutions has been shown. The hit-rate comparison has also shown that their scheduling model produces a higher hit-rate than the random scheduling model, even though with the same limitation pointed in the delay evaluation.

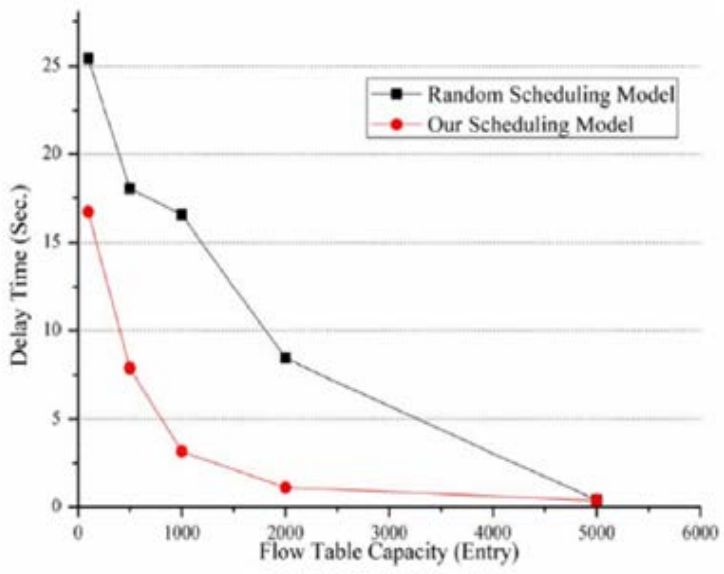

Figure 2: Delay time comparison under two policies

2.1.2 Design and Evaluation of Learning Algorithms for Dynamic Resource Management in Virtual Networks

In the context of virtual networks, [4] studied the Virtual Network Embedding (VNE) problem, on which virtual nodes are embedded to resources in the underlying physical network. In all solutions studied, even though there is some kind of dynamic resource allocation, part of the physical network is still tied to each of the virtual nodes. This prevents that these resources get fully migrated to nodes within peak usage when this virtual node is idle. They proposed a distributed dynamic resource allocation model using an AI technique called reinforcement learning and an initialization scheme to improve the rate of convergence in the allocation. In the paper, a virtual network (VN) is represented by a weighted undirected graph whose vertices are virtual nodes and its edges are virtual links. In a virtual network graph, each virtual node has three properties: position, queue size and maximum position deviation. Each virtual link connecting a vertex i to a vertex j has two properties: maximum packet delay and allocated bandwidth. A substrate network (SN) consists in the physical network resources to be

allocated, it is also modeled by a weighted undirected graph whose vertices are physical nodes and the edges are physical links between nodes.

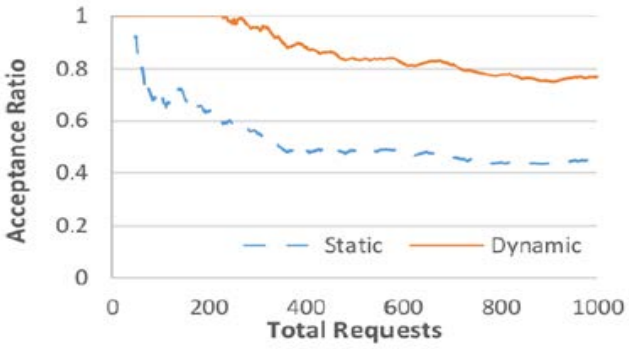

Figure 3: VN Acceptance Ratio

The Q-learning algorithm has been used by the authors to map the network model within the reinforcement learning. This learning algorithm temporally builds information about the next possible state to be taken. This is done via policies that determine the behavior of the learning agent. Q-values Q(s,a) consist in the next states that will be taken based on action a. In the proposed learning method, the actions of decreasing, increasing or maintaining the allocation of resources are based in the states of resources already allocated and in use, the ones allocated and idle and the ones free in the substrate network. The policy initialization relies on initializing the states that easily represent the expected actions of the agents in order to minimize the converting time of the algorithm, hence producing faster optimal resource allocation. Results of simulations on NS-3 have shown that the algorithm successfully enables a better virtual network acceptance ratio because of the higher availability of SN resources (Figure 3). On the other hand, Figure 4 shows that the packet drop ratio in the learning algorithm is higher due to learning phase of the agents. In this case, the trade-off between the learning phase and the long-term resource allocation needs to be identified in order to ensure that the algorithm is properly configured and is valuable for the whole system.

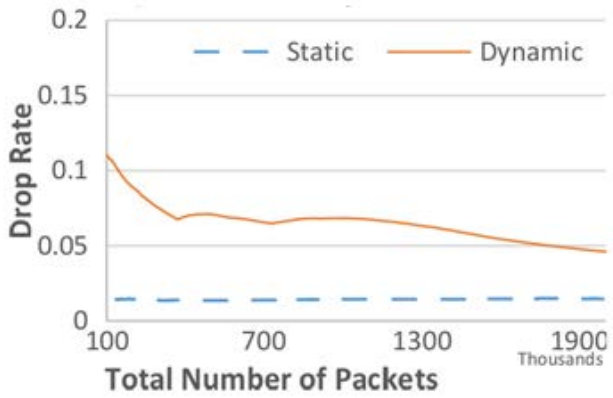

Figure 4: Node Packet Drop Rate

2.1.3 Hierarchical Network-Aware Placement of Service Oriented Applications in Clouds

Reference [5] Proposed a model based on Integer Linear Programming (ILP) for the Cloud Application Placement Problem (CAPP). CAPP is used to determine how applications and services are allocated within the cloud. For instance, in which machine should a service be allocated in order to satisfy multiple constraints such as CPU, memory, bandwidth and management policies? The definition of application in this work applies only to those designed with a service-oriented architecture (SOA), on which the requirements of each application are achieved using a set of communicating services. Considering the scenario of cloud providers, the main goal of dynamic resource allocation is to use the least amount of resources, e.g. using less computing nodes, while also ensuring that overall application performance satisfies customer's needs. Optimal solutions for CAPP can be obtained, but are not scalable since the problem is NP-Hard. In order to keep the management system scalable and also maintain an acceptable level of knowledge about the global system state, authors choose to use a hierarchical-based approach.

They have proposed i) a formal model for a network-aware CAPP designed to deal with applications built using the SOA architecture, ii) an ILP-based hierarchical algorithm to optimize the diversity of parameters in the network-aware CAPP, iii) a comparison between the optimal ILP algorithm with two hierarchical algorithms based on Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) that find near-optimal solutions for the CAPP problem. Optimization objectives defined in this paper are i) maximization of accepted requests, ii) network satisfaction and iii) minimization of the a) number of computing nodes used, b) number of service migrations and c) hop count between communicating services. The results presented have compared the models based on PSO and GA with an optimal ILP-based algorithm (Figure 5). Even though their algorithms can accept 8% less requests than the ILP-based one, their proposal can execute 100 times faster than the ILP-based solution when scalability is taken in consideration.

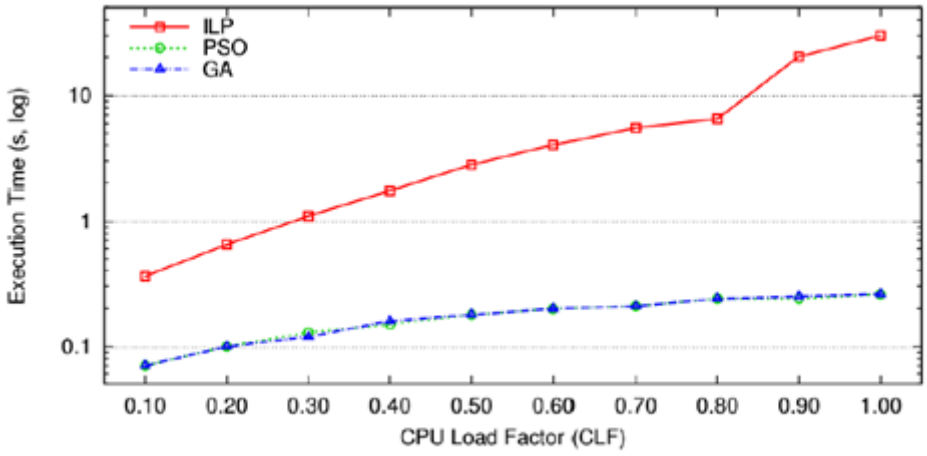

Figure 5: Comparison of the execution speed of the algorithms

2.1.4 Integrating Cloud Application Auto scaling with Dynamic VM Allocation [8] Proposed an approach that combines both Dynamic VM Allocation with Cloud Application Auto scaling in order to reduce operational

costs to cloud providers and clients. Cloud providers are the owners of the infrastructure and clients are the ones that use the underlying infrastructure to deliver applications to end users, while also ensuring that Service Level Agreements (SLAs) are met. The authors have chosen to work on cloud computing focused to the Infrastructure as a Service (IaaS), on which clients have low-level access to the VMs. In this approach, clients may deploy, manage and scale applications themselves in an on-demand and pay-per-use form. In order to exploit the availability of resources and scaling capabilities of the cloud, applications must dynamically scale on demand. In this work, an application is considered as a set of communicating VMs providing a transparent and single-faced service to the end users.

Resource provisioning, in the client perspective, must allow dynamic addition or removal of resources to both minimize operational costs by avoiding overprovisioning under average or smaller loads, and meet performance goals under peak loads. In the cloud provider perspective, it is important to offers its services to more clients and consolidate as much load as possible with the least amount of physical resources to reduce energy and hardware costs. These cost reductions are achieved mainly through resource over committing, for instance, by allocating more virtual CPUs to VMs than physical CPUs available in a server. However, over committing may lead to resource contention and live VM migrations must be performed dynamically to maintain an acceptable performance for clients' applications. Dynamic VM allocation is, therefore, the process of periodically reallocating VMs in response to constantly changing resource requirements. Related works have focused mainly in either Dynamic VM Allocation or in Cloud Application Auto scaling using static approaches, linear programming and best-fit heuristics.

An Application is divided into one or more tasks, which are then divided into task instances. In this work, the authors consider only interactive applications and evaluate the approach through simulations in the DCSim simulator. Application performance is evaluated by the SLA. In this context, SLA is defined as an upper threshold in the response time. Application Auto scaling is achieved through a heuristic and rule-based auto scaling algorithm. Every application has its own manager that collects data from every task instance in regular intervals. This algorithm scales up applications based on their response time and scales down based on their CPU utilization when the application response time is below a warning level, which is a percentage of the SLA response time threshold, in a sliding window. Dynamic VM Allocation is achieved through classification of hosts in the following categories i) stressed hosts, i.e. with higher response times; ii) partially utilized hosts, iii) underutilized hosts and iv) empty hosts.

The approach consists in three operations with VMs: i) Relocation, ii) Consolidation and iii) Placement. In Relocation, the four categories of hosts are sorted in order of CPU utilization and stressed hosts are used as a source of VMs for migration until the CPU usage is under a defined threshold. Consolidation consists in picking hosts with low resource usage, migrating its VMs to another host and then switching the host into a lower power state. Placement selects a host for instantiating a new VM, similar to VM Relocation. These two approaches are combined into a single algorithm and later in the paper the authors evaluate its results. Evaluation of the algorithm is achieved through a comparison between static resource allocation, considering only Auto scaling, then combining it with Dynamic VM Allocation and finally with the integrated algorithm (Table 1). Results obtained in the simulations for Auto scaling suggest a reduction in operational costs by reducing the amount of

resources used to 64%, while slightly reducing the SLA from 100% (static allocation) to 95.8%. Combining Auto scaling with Dynamic VM Allocation also reduces power consumption but increases the number of migrations. The integrated algorithm proposed in this work shows, through the simulation results on Table 1, that SLA achievement, time in which the response time SLA was under the upper threshold, has increased and, furthermore, the number of migrations decreased. However, this evaluation lacks comparison with other approaches since the algorithm for dynamic relocation was developed by the same authors and it is also limited to static resource allocation. In summary, the context of interactive applications considered in the paper is very specific, although important for many public cloud providers.

Table 1: Algorithm Comparison

Algorithm	Hosts	Power	SLA	AS Ops	Migrations
Static	194.7	4177kWh	100.0%	N/A	N/A
Autoscaling	147.7	3424kWh	95.8%	9503	N/A
Separate	80.6	2066kWh	90.7%	11895	21834
Integrated	100.7	2460kWh	95.3%	10320	3778

3. Conclusion and Next steps

The daily development of enhanced features over Cloud computing and Virtual Machines has made its management a challenging task. The growing user demands require that the underlying resources are available on-the-fly at any time. Nevertheless, the manual management of resources is unreliable and unfeasible to scalability and availability. Software Defined and Virtual Networks have gained space as a proper solution to the management of resources through the dynamic resource allocation. This is possible due to the decoupling and abstraction of the network infrastructure, making the networks programmable by means of high-level policies. In this partial paper, we have surveyed some of the main approaches to manage the dynamic resource allocation. Reference [2] Introduce the integrated allocation of link bandwidth and flow table for multiple control applications in SDN. [4] Create a machine learning based approach to dynamic resource management in virtual networks. Reference [5] Present hierarchical bio-inspired algorithms to solve the Cloud Application Placement Problem. Reference [8] Propose an algorithm that integrates automatic scaling of cloud applications with dynamic allocation of virtual machines. Our goal in the full paper is providing a comparative analysis of these solutions in order to identify the main criteria already studied and point fields to explore.

References

[1] Houda Jmila, Djamal Zeghlache (2015). An Adaptive Load Balancing Scheme for Evolving Virtual Networks, 2015 IEEE12th Annual IEEE Consumer Communications and Networking Conference (CCNC)

[2] Chowdhury, N. and Boutaba, R. (2010). A survey of network virtualization. Computer Networks, 54(5):862–876.

[3] Feng, T., Bi, J., and Wang, K. (2014). Joint allocation and scheduling of network resource for multiple

control applications in SDN.

[4] Hu, F., Hao, Q., and Bao, K. (2014). A survey on software defined networking (SDN) and open flow: From concept to implementation.

[5] Mijumbi, R., Gorricho, J.-L., Serrat, J., Claeys, M., De Turck, F., and Latr´e, S. (2014). Design and evaluation of learning algorithms for dynamic resource management in virtual networks. In IEEE/IFIP Network Operations and Management Symposium (NOMS), Krakow, Poland.

[6] Sreenivasa Reddy, D Venkatesh, K Ramesh (2015). Consignment Re-Balancing for Dispersed File Schema in Clouds. In International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 6, June 2015

[7] Moens, H., Hanssens, B., Dhoedt, B., and De Turck, F. (2014). Hierarchical network aware placement of service oriented applications in clouds. In Network Operations and Management Symposium (NOMS), 2014 IEEE, pages 1–8. IEEE.

[8] Nunes, B., Mendonca, M., Nguyen, X., Obraczka, K., and Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks.

[9] Parikh, S. M. (2013). A survey on cloud computing resource allocation techniques. In Engineering (NUiCONE), 2013 Nirma University International Conference on, pages 1–5. IEEE.