GreenCloud for Simulating QoS-based NaaS in Cloud Computing

Jiang Zhihua
Mathematics & Information Science Academy
Leshan Normal University
Leshan City, Sichuan Province, P.R.China
lai027@163.com

***Abstract*—Cloud computing that was introduced when local facilities could not any more satisfy access demands along with the increase of network bandwidth and diversity of services, breaks constraints of physical deployment and delivers varieties of services in its hardware/software virtualized manner. Being inherited the idea, the NaaS as a new presented research field ever since the cloud computing being introduced, delivers network itself as a kind of service under such circumstances. By its concept, the network utilization will no longer be restricted by physical network or local facilities, but is able to expropriate any network resources within the cloud once they are virtualization organized. GreenCloud simulation as one of the NS-2 cloud computing simulators, implements an energy-aware cloud computing scenario with infrastructures (switches, routers, links, etc.) virtualized in datacenter. In this paper, we discusses the Network as a Service (NaaS), a cloud architecture that delivers virtualized network as per demands from cloud user, with illustration of QoS-based GreenCloud simulation, that to present the behavior and performance of NaaS in cloud computing environment.**

Keywords-cloud computing; NaaS; QoS; GreenCloud; simulation

I. INTRODUCTION

When various kinds of video slices are systematically posted onto the YouTube for sharing, or fuzzy voice search is provided by the Google App for one who may lost navigation, people who benefits from conveniences and simplicities brought by those services is barely aware of the existence of computing. From contrary perspective concerning however, ignore of 'backyard' computing is the very goal that network service pursues throughout its entire development. Computing is more like a strange and sophisticated existence hidden itself behind the back of network infrastructure, delivering varieties of services through the network, even the network itself as one of them.

Nevertheless, focuses has been gathered again when the cloud concept has been presented since new century. The cloud computing is a new paradigm based on internet that it provides on-demand computing for individuals or enterprises through hierarchic services. Cloud computing belongs to an advantaged development of distributed computing, parallel computing and grid computing. Since there is still no strict definition to could computing, it can be considered from both commercial and technical points [9][10]. Its most attracting fact is that users save their cost on hardware purchase, but to 'pay-as-per-use', fetch services they need from 'sliced' services, those sliced services can be software (Software as a Service, i.e., SaaS), network infrastructure (Infrastructure as a Service, i.e., IaaS), application platform (Platform as a Service, i.e., PaaS) and so on [3]. From such perspective, the cloud computing can be considered as commercial implementation out of computer science. Most importantly however, since strength of fundamental facilities deployed in network topology does not match ambitions of bandwidth and application, cloud computing can be a break through that every local unit dedicates itself as element (cloudlet) in the cloud environment. Meanwhile, every single user is able to take services by utilizing the integrated infrastructure (Data Center, i.e., DC) without restrain of local hardware/software. Cloud computing operates its computing all over huge amount of distributed computers of the network instead of local computer or remote server, which turns out that the Data Center (DC) acts more like virtual internet role rather than isolated element so that resources can be shifted on-demand and huge network resources can be efficiently applied.

Generally, the idea of NaaS allows the operator offers varieties of cloud-based network services for users, including routing, forwarding, QoS and network addressing and so on. All those telecommunication attributes as services are implemented depending on the internet itself that the user (i.e., tenants by the concept of cloud network) does not have to prepare any additional topology resources. The NaaS is to make the 'black-box' that was holding by telecommunication provider more transparent, that once demanded, tenants are able to control and manage network performance in order to deliver more efficient service, in this case, the network with its performance can be considered as a service to the tenants [1].

Thus, firstly, the NaaS is completely based on cloud environment and secondly, the virtualization to the networks must be covered and combined all over the cloud. As matter of fact, the idea of NaaS is to create a virtual network upon many of physical network infrastructures.

In this case, NaaS consumption and NaaS QoS would become significant, because such a virtualized network will confront problems of integration and compatibility among those physical networks. It is expected that issues of power consumption and QoS performance needs to be well addressed during the NaaS research.

GreenCloud simulator implements energy-aware cloud computing scenarios with almost all infrastructure elements virtualized in datacenter and it manipulates a workload-orientated simulating mechanism that the main objective it measures is power consumption and cost of infrastructure under cloud computing environment [4][5]. Besides, with the support of QoSbox embedded in the simulator, the GreenCloud is able to provide QoS ability when NaaS is delivered through the cloud network topology. This paper is

978-1-4799-2548-3/13 $31.00 © 2013 IEEE
DOI 10.1109/CIS.2013.167

with assistant of the GreenCloud, to implement a simulated NaaS structure under a QoS guaranteed cloud environment and evaluate the NaaS attributes including its performances [6].

II. Related Work

Different from traditional network that establishes connections of physical node to branch, server to server/client and uplink aggregation to downlink access, etc., the NaaS aims at creating virtual network at all manners with application supportive. Physical networks in this case are considered as resources integrated in NaaS. In order to meet the principle of service, any connection inside the NaaS mechanism must be established on-demand, i.e., temporarily rather than being provisioned in advance. Additionally, in traditional network, applications being installed inside network terminals inherit network connectivity and behavior; when it comes to NaaS, applications themselves are able to designate connectivity and performance requirement they need, rather than acclimatize themselves to network environment. Such subjectivity allows the feasibility of connectivity and performance requirement being able to provide to application on request, which on the contrary allows user determination of whether the NaaS is the best choice.

Question is, and perhaps will be for quite a while, how to let NaaS deployment become practical available. First, the NaaS needs a particular agreement as collaborating relationship between application and network; next, it requires a specified mechanism to create such a demand-response relationship. In order to implement the QoS-based NaaS under a cloud environment via GreenCloud simulator, the following features will be introduced to clarify such collaboration, as well as the instruction of relative simulation facilities.

A. QoS in Cloud Environment

A basic QoS model based on cloud environment is shown as figure below in Figure 1. :

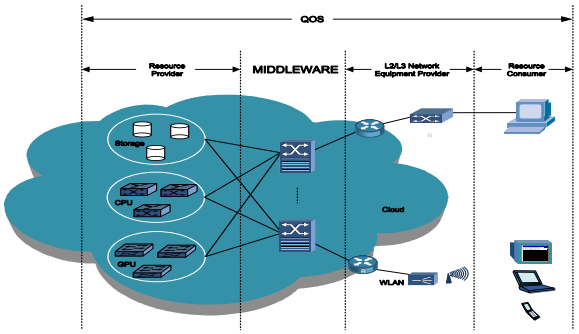

Figure 1. QoS Model in a Cloud Environment

QoS cloud environment consists of four general roles. Resource provider provides (such as Google cloud, Amazon AWS and Microsoft Windows Azure, etc. [2]) services or resources including multimedia (storage, computing, data processing, software environment). Resource consumer plays the roles of resource user and renter to serve their own requirements. Middleware provides connectivity between resource provider and resource consumer and virtualizes varieties of resources to resource pool via virtualization technologies, so that those resources can be provided to the consumer. Additionally, the middleware conducts user management, task management, resource management and security management to provide reliable quality guarantee to user services and task operation, especially to meet the requirement of diversity in multimedia cloud environment.

Due to the resources provided to user in the QoS cloud environment model are provided as resource pool, Virtual Machines (VM) must be constructed at each resource type that they are able to offer application to one or more user simultaneity as if every single user is occupying computing resource alone. In the following resource reservation model (Figure 2.), new resource information will be forwarded and stored to the proxy via resource server. Once a new cloud task is submitted to cloud platform, the task analyzer recognizes its QoS requirement first and then forwarded the requirement to the proxy who is able to match it with resource information and see if they meet each other [7]. In the end, the resource server virtualizes cloudlet according to the resource being confirmed by the proxy.

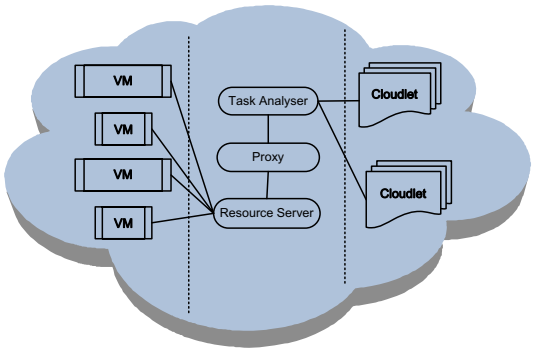

Figure 2. Cloud-based Resource Reservation Model

B. QoSbox in GreenCloud

Different from other QoS architectures, The QoSbox is actually belonging to ‘a configurable IP router that provides per-hop service guarantees on loss, delays and throughput to classes of traffic’ [12]. The QoSbox does not ‘rely on any external component (e.g., no traffic shaping and no admission control) to enforce the desired service guarantees, but instead, dynamically adapts packet forwarding and dropping decisions as a function of the instantaneous traffic arrivals’. In addition, the QoSbox can ‘enforce both absolute bounds and proportional service guarantees on queuing delays, loss rates, and throughput at the same time’ [12].

Programmed by open source C++ and Otcl, the GreenCloud simulator along with QoSbox inside, is based on NS-2 simulation environment, which belongs to the one of objective-oriented, discrete event driven simulation. The Joint Buffer Management and Scheduling (JoBS) as the key implementation in NS-2 whose objective is providing

absolute and relative loss rate ratio and independent delay differentiation for each node class, 'JoBS' algorithm thus ensures services for each hop during the data transmission. The representation requires constraints from QoS and as an instance, the QoS as the constraint can be applied as the following three issues:

- Class1 Delay ≈ 2 * Class2 Delay
- Class2 Delay ≈ 10(-1) * Class3 LossRate Ratio
- Class3 Delay ≦ 5 ms

The former 2 of the above 3 instances are belonging to relative constraints. The constraints can be considered as mixture of any absolute and relative constraints and in particular, 'JoBS' supports 5 types of constraints in total:

- Relative Delay Constraint (RDC) indicating proportional delay differentiation between classes, e.g., as for Class1 and Class2 as above, the relationship would be:

$$(\text{Class2 Delay})/(\text{Class1Delay}) \approx \text{constant number} \quad (1)$$

- Absolute Delay Constraint (ADC) indicating that ADC at ClassN requires the delay at ClassN must satisfies the delay value with no worse than DelayN (d_N);
- Related LossRate Constraint (RLC) indicating proportional loss-rate differentiation between classes;
- Absolute LossRate Constraint (ALC) indicating that ALC at ClassN requires the delay at ClassN must satisties that delay value with no exceeding than the threshold LossRateN(L_N);
- Absolute Rate Constraint (ARC) indicating that the ARC at ClassN requires that output of ClassN must have a bottom value as μ_N.

The priority for QoS constraint in 'JoBS' algorithm is shown as follows:

- ALC> ADC
- ARC>Relative Constraints

The priority mechanism indicates that if the 'JoBS' cannot satisfy absolute and relative constraints at the same time, the ADC will be with highest consideration.

III. IMPLEMENTATION OF QOS-BASED CLOUD NAAS

A. *GreenCloud Simulation*

The GreenCloud simulator (the 'GreenCloud Virtual Machine' can be fetched through http://greencloud.gforge.uni.lu/index.html) can be operated under a VM environment (taking the VMware workstation as an example). Via folder 'home/greencloud/src/scripts/', network infrastructure can be deployed through file 'topology.tcl' which is for modification of datacenter topology, shown as Figure 3. below:

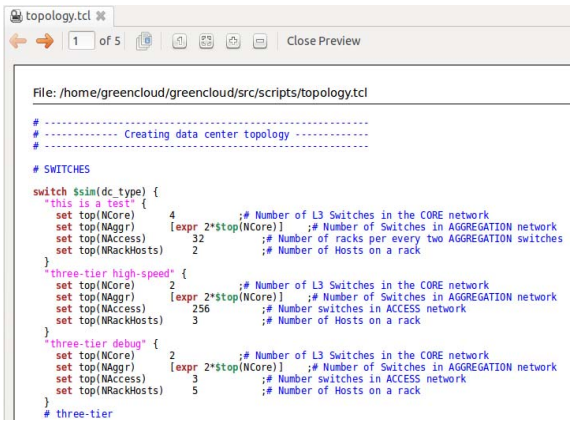

Figure 3. Deployment of Data Center Topology

The simulation will last spend default duration (60 seconds, configurable) after configuration is done. Figure 4. below shows the result with power consumptions, etc.:

Figure 4. GreenCloud Simulation Results

B. *QoSbox Simulation*

In order to evaluate the performance when NaaS is delivered as a service in cloud computing, QoS guarantee would be one of the most significant features that whether and how the NaaS satisfies QoS demands of diversity of user requirement, the implementation of QoSbox can be potential if GreenCloud is applied for cloud simulation.

The main objective of the simulation is to implement QoS guaranteed GreenCloud simulation by deployment of QoSbox into the simulation and evaluate if attributes of QoS can be guaranteed and delivered under proper cloud computing scenarios, goals may be including:

- QoSbox Utilization in GreenCloud Simulation;

- To change network topology such as user number, and see if there is influence to network performance;
- To designate different service type for different user and see if there is influence to network performance;

Simulation of QoSbox is achieved through the provision of file 'jobs-lossdel.tcl' out of category '/home/greencloud/greencloud/build/ns-2.35/tcl/ex/jobs', result can be shown in animation via executing command:

```
ns jobs-lossdel.tcl                                    (2)
```

The animation shows how QoSbox respectively manages traffic flows with loss rate per flow, referring to Figure 5. below:

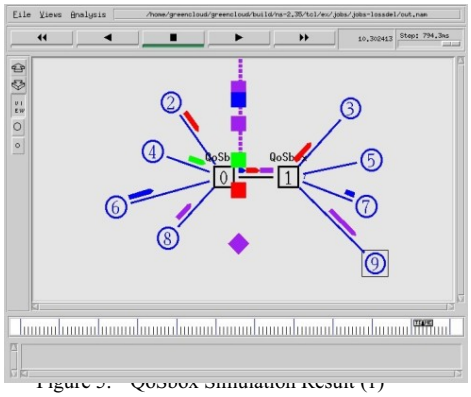

Figure 5. QoSbox Simulation Result (1)

As to the QoSbox simulation with above topology deployment, descriptions are as follows:

- the topology shows single QoSbox scenario that 2 PCs are acting as routers with QoSbox mechanism and another 8 PCs are acting as terminal nodes;

```
# jobs-lossdel.tcl
#
# Tests that the delay/loss differentiation is working
properly
# Outputs a nam animation and tracefiles.
#
# Topology is as follows:
#
# source(1)  --\                                  /- sink(1)
# source(2)  ---\  core_node(1) -- core_node(2)  /-- sink(2)
# source(3)  ---/                                \-- sink(3)
# source(4)  __/                                  \_ sink(4)
#
# Links:
# - : 10 Mbps, 1 ms
#
# Classes:
# 1: ADC=5 ms, ALC=1 %, no ARC, no RDC, no RLC
# 2: no ADC, no ALC, RDC = 4, RLC = 2
# 3: no ADC, no ALC, RDC = 4, RLC = 2
# 4: no ADC, no ALC, RDC = 4, RLC = 2
#
# Traffic is randomized CBR
#
# Auxiliary functions
```

Also, the file 'jobs-lossdel.tcl' describes how the topology organizes:

```
# nam stuff
$ns duplex-link-op $source(1) $core_node(1) orient 300deg
$ns duplex-link-op $source(1) $core_node(1) color "blue"
$ns duplex-link-op $source(2) $core_node(1) orient 330deg
$ns duplex-link-op $source(2) $core_node(1) color "blue"
$ns duplex-link-op $source(3) $core_node(1) orient 30deg
$ns duplex-link-op $source(3) $core_node(1) color "blue"
$ns duplex-link-op $source(4) $core_node(1) orient 60deg
$ns duplex-link-op $source(4) $core_node(1) color "blue"
$ns duplex-link-op $core_node(1) $core_node(2) orient right
$ns duplex-link-op $core_node(1) $core_node(2) color black
$ns duplex-link-op $core_node(1) $core_node(2) queuePos 0.5
$ns duplex-link-op $core_node(2) $sink(1) orient 60deg
$ns duplex-link-op $core_node(2) $sink(1) color "blue"
$ns duplex-link-op $core_node(2) $sink(2) orient 30deg
$ns duplex-link-op $core_node(2) $sink(2) color "blue"
$ns duplex-link-op $core_node(2) $sink(3) orient 330deg
$ns duplex-link-op $core_node(2) $sink(3) color "blue"
$ns duplex-link-op $core_node(2) $sink(4) orient 300deg
$ns duplex-link-op $core_node(2) $sink(4) color "blue"
$ns color 1 red
$ns color 2 green
$ns color 3 blue
$ns color 4 purple
```

- QoSbox simulation is generated from Router 0 and goes through Router 1 (referring to Figure 5.), with management over 4 flows from left to the right;
- One single flow reflects a source (marked as even node) and a sink (marked as odd node) at two terminals and the data flow is from source to sink. There are 4 different colors representing 4 flows in the figure, e.g., flows from Node 2 to 3, 4 to 5, 6 to 7 and 8 to 9;

```
# Number of monitored flows (equal to # of classes)
set N_CL       4
puts "Number of classes: $N_CL"

set N_USERS    4
puts "Number of flows: $N_USERS"

set START_TM   0.0
puts "Monitored flows start at: $START_TM"

set max_time   10.0
puts "Max time: $max_time (sec)"
```

- For each flow, there can be more than one traffic class, which means multiple classes (service type) can be deployed at a single flow;

```
# Classes:
# 1: ADC=5 ms, ALC=1 %, no ARC, no RDC, no RLC
# 2: no ADC, no ALC, RDC = 4, RLC = 2
# 3: no ADC, no ALC, RDC = 4, RLC = 2
# 4: no ADC, no ALC, RDC = 4, RLC = 2
```

- Flow number can be modified along with its class number, indicating both of the user number (assuming each node no matter the node is source or sink, can be considered as user PC) and service type can be changed.

```
# Number of monitored flows (equal to # of classes)
set N_CL       4
puts "Number of classes: $N_CL"

set N_USERS    4
puts "Number of flows: $N_USERS"
```

Parameters in above text can be modified as per simulation requirement. Now assuming:

```
# Number of monitored flows (equal to # of classes)
set N_CL       2
puts "Number of classes: $N_CL"

set N_USERS    2
puts "Number of flows: $N_USERS"
```

```
# nam stuff
$ns duplex-link-op $source(1) $core_node(1) orient 300deg
$ns duplex-link-op $source(1) $core_node(1) color "blue"
$ns duplex-link-op $source(2) $core_node(1) orient 330deg
$ns duplex-link-op $source(2) $core_node(1) color "blue"
$ns duplex-link-op $core_node(1) $core_node(2) orient right
$ns duplex-link-op $core_node(1) $core_node(2) color black
$ns duplex-link-op $core_node(1) $core_node(2) queuePos 0.5
```

```
$ns duplex-link-op $core_node(2) $sink(1) orient 60deg
$ns duplex-link-op $core_node(2) $sink(1) color "blue"
$ns duplex-link-op $core_node(2) $sink(2) orient 30deg
$ns duplex-link-op $core_node(2) $sink(2) color "blue"
$ns color 1 red
$ns color 2 green
```

```
# Classes:
# 1: ADC=5 ms, ALC=1 %, no ARC, no RDC, no RLC
# 2: no ADC, no ALC, RDC = 4, RLC = 2
```

The topology will be changed as shown as in following figure (Figure 6.):

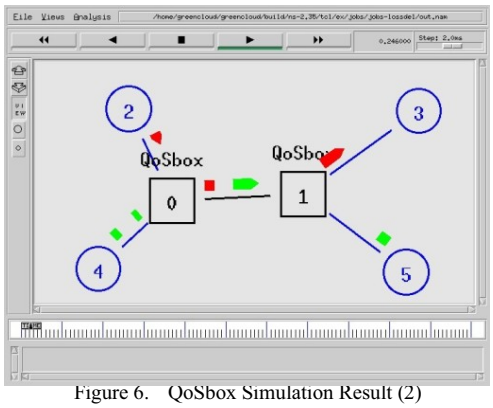

Figure 6. QoSbox Simulation Result (2)

Additionally, the 'jobs-lossdel.tcl' allows the simulation to designate QoS factors such as bandwidth, delay, throughput and jitter (reflecting buffer size), referring to text below:

```
puts "\nTOPOLOGY SETUP"

# Number of hops (=k)
set hops 2
puts "Number of hops: $hops"

# Link  latency (ms)
set DELAY      1.0
puts "Links latency: $DELAY (ms)"

# links: bandwidth (kbps)
set BW                  10000.0
puts "Links capacity: $BW (kbps)"

# Buffer size in gateways (in packets)
set GW_BUFF    50
puts "Gateway Buffer Size: $GW_BUFF (packs)"

# Packet Size (in bytes); assume common for all sources
set PKTSZ      500
set MAXWIN     50
puts "Packet Size: $PKTSZ (bytes)"
```

As it was mentioned that the key difference between the QoSbox and other QoS architectures is that the QoSbox does not rely on external mechanism or component to enforce the desired service guarantees. 'The main limitation of an approach dynamically adapting to the instantaneous traffic arrival is that, incases caused for instance by a sudden large burst of traffic, it may be impossible to satisfy all delay bounds and loss rate bounds at the same time [8]. In such cases, some bounds have be temporarily relaxed. Thus, an order of relaxation of the service guarantees must be chosen at design time.' [12]

IV. Conclusion

The significance and potentiality of NaaS development can be mediate that the solution enhances network security and enlarges operation simplicity because of QoS guarantee and high level of application integration; on the other hand, it can be direct that the utilization of NaaS is able to instantly hook internet resource payment [11]. Although more and more researchers are considering that the NaaS brings flexible new network service field, the virtualization of NaaS has limit that currently it only distributes accesses providing services to practical network however. As discussed, the potential issue is that how to improve the compatibility of control software in cloud, for virtualized NaaS has not yet implemented real integration with network infrastructures, which brings difficulties of management. Even worse, if virtual network grows without aware of influences from its increasing traffic flow, QoS problem will be brought to all network users by such a virtualization and that of device-based NaaS.

References

[1] Paolo Costa, Matteo Migliavacca, Peter Pietzuch, Alexander L.Wolf, "NaaS: Network-as-a-Service in the Cloud", Imperial College London, University of Kent, 2008, pp.13–145.

[2] VARIA, J., "Cloud Architectures. Amazon Web Services", 2009.

[3] Christian Baun, Marcel Kunze, Jens Nimis, Stefan Tai, "Cloud Computing—Web-Based Dynamic IT Services", Springer Herdelberg Dordrecht London New York, 2011, pp.243–255.

[4] Baris Aksanli, Jagannathan Venkatesh, and Tajana Šimunic´ Rosing, "Using Datacenter Simulation to Evaluate Green", University of California, San Diego, 2007, pp.25.

[5] D. Kliazovich, P. Bouvry, and S. U. Khan, "A Packet-level Simulator of Energy-aware Cloud Computing Data CentersEnergy Integration", University of Luxembourg.
http://greencloud.gforge.uni.lu/ftp/greencloud.pdf.

[6] Saurabh Kumar Garg and Rajkumar Buyya, "*Green Cloud computing and Environmental Sustainability*", University of Melbourne, 2011.
http://www.cloudbus.org/papers/Cloud-EnvSustainability2011.pdf

[7] Y.S.Dai,M.Xie, K.L.Poh, "A Study of Service Reliability and Availability for Distributed System [J]. Reliability Eng and System Safety [J]", 2003,79 (1), pp.103-112.

[8] Li ZD, Xie L, "Research on Ensuring QoS and its Admission Control in Web Servers [J]. Journal of Computer Research and Development", 2005,42 (4), pp.662-668.

[9] Virtualization & Cloud Computing Group, "Virtualization and Cloud Computing [M]", Beijing Electronic Industry Publish. 2009: 127-135.

[10] Armbrust M, Fox A, Griffith R, "Above the Clouds: A Berkeley View of Cloud Computing [J]", 2009. 2(23):, pp.45-48.

[11] Dogan A, Ozguner F (2002), "Scheduling Independent Tasks with QoS Requirements in Grid Computing with Time-varying Resource Prices [J]", In: CCGRID 2002, pp. 58-69.

[12] Nicolas Christin J¨org Liebeherr, Department of Computer Science, "The QoSbox: A PC-Router for Quantitative Service Differentiation in IP Networks", University of Virginia, 2009, pp.7-23.