Load-Balancing for Edge QoE-based VNF placement for OTT Video Streaming

Utku Bulkan, Muddesar Iqbal, Tasos Dagiuklas,

Division of Computer Science, London South Bank University, UK

Email: {bulkanu, m.iqbal ,tdagiuklas}@lsbu.ac.uk

***Abstract*—Over The Top (OTT) service providers require platforms to support distributed, complex, cloud-oriented, scalable, micro-service based systems. Such systems require on-the-fly placement of Virtual Network Functions (VNF) to support streaming and transcoding of content based on QoE feedback provided by the end-user. This paper proposes a QoE Scheme to support on-the-fly virtual network functions deployment for OTT video streaming and transcoding. The QoE feedback considers limited cloud resources, transcoding requirements, throughput and latency. Both horizontal and vertical scaling strategies (including VM migration) are discussed to cover up availability and reliability of intermediate and edge Content Delivery Network (CDN) cache nodes.**

***Index Terms*—QoE, Cloud, VNF, OTT Video Streaming**

I. Introduction

ONLINE video market has been growing exponentially over the last decade. Globally, IP video traffic will be 82 percent of Internet traffic by 2021 [1]. Internet video will continue to grow at a rapid pace. Additional demand necessitates a parallel advance in scalability, availability and reliability requirements. Depending on the system implementation, it is generally quite easy to meet these demands by running more Virtual Machine (VM) instances [2]. However, this might trigger a corresponding increase in cloud hosting costs [3]. Since the introduction of Content Delivery Networks (CDN) [4], the architecture of video delivery systems have evolved. This has led to a breakthrough in efficiency by many aspects, including service capacity, reduced latency and better cache management [5]. The procedure starts with the user request, followed by a pull model [6] caching unless a pre-push model [7] is configured [8]. The content that is frequently used stay in the cache longer time [9]. Depending on different CDN deployments, the distributed cache nodes may have the capability to search other nodes' caches [10] for a requested content and copy it from a closer and cost-efficient neighboring node. Current academic research viewpoint [5,7] and state-of-the-art technology point of view [3, 8, 9, 11] provide an understanding that only relies on objective network metrics and cloud resource constraints whereas this paper introduces a brand new foundational understanding of the impact of QoE on load balancing and resource optimization. The edge CDN cache pulls the content, and end-users get the service via their video players [12].

The architecture of the working mechanism of edge content nodes [13] involves cache content copy that resides in a VM as VNF that is pulled from origin [14]. Actual contact points for the users are the front-line load balancers [15] that redirect requests to the containers that run web servers [16], which deliver the chunks of video data. Therefore, optimizing the number of running VM instances in the cloud [17], plays a crucial role for enhancing the QoE. Any unexpected peak in user requests results in a parallel-unforeseen scalability demand and equivalent unpredicted costs on the cloud. An attempt to confront this demand requires other additional investment on redundancy [3].

The primary intention of this paper is to overcome the limitations of the solutions that have been proposed in the literature discussed above, taking into account video QoE characteristics. For this purpose, scalable online video delivery systems have been developed to compare the proposed QoE-based scheme against different load balancing strategies [18] to provide an on-the-fly orchestration to rebalance the limited cloud resources [19]..

The remainder of the paper is organized as follows: Section II discusses related works and literature review. Section III presents various types of scaling algorithms. Section IV introduces a proposed QoE-based scheme for VNF placement. Section V explains warming up and cooling down machanisims and compares the performance of scaling strategies. Section VI formulates computational resource constraints for online video streaming via VNFs. Finally, Section VII concludes with the results and future work.

II. Related Work

Defining a scalable methodology for cloud-based services has attracted a lot attention due to the demand for distributed applications that provide reliability [20], durability [21] and availability [22]. Kesevaraja et al have modeled [23] single VM instance taking into account the success rate of a physical node, current utilization of the processor, the maximum capacity of the processor,, current utilization of primary memory, maximum available capacity of primary memory, the data bits transferred among time interval and the network bandwidth. Chunlin et al have proposed multiple context-based service-scheduling models [24] that adopt network utility maximization framework to maximize total system utility. When the mobile device application's job is accepted by the cloud system, it is scheduled and assigned to the cloud resource. Bilal et al have provided a formula [25] for cloud costs taking into account computational instances, the total

amount of data in bps required for the server and the total cost for data per second.

QoE for OTT has been standardized in ITU-T P.1203.3 recommendation [26]. In this recommendation, a media session quality score is formulated based on a number of stalls, total stall duration, buffering duration. This provides a basis for a single user's watching experience.

$$\text{QoE} = e^{-\frac{numStalls}{s1}} . e^{-\frac{\left(\frac{totalbuflen}{T}\right)}{s2}} . e^{-\frac{\left(\frac{bufdur}{T}\right)}{s3}} \quad (1)$$

This paper proposes a hybrid scalability model for VNF placement that considers QoE, cost, and resource efficiency aspects of online video delivery a dynamic virtual network function with on-the-fly deployment in multi-location cloud based on the QoE feedback. Comparison of pros & cons for different scaling strategies is presented. Additionally, formalization of memory and computation demand related to video parameters clarifies the usage of cloud instances with real-life scenarios on cloud [27].

III. Load Balancing Strategies for Edge Cloud Computing

There are several load balancing strategies widely employed in web-based services based on random-access, number of users, throughput, CPU usage, memory efficient. In this section, these strategies are going to be -presented.

A. Random-access (a.k.a. Round-Robin)

Random-access load balancing works on the assumption that the users should connect randomly to any server through a list of available servers. The definition of randomness becomes an important fact and strictly related to the expected number of users that intend to use the service [28].

$$\text{u}_A(\text{t}) = \frac{\sum_{i=0}^{v} u_i(t)}{v} \quad (2)$$

Average number of users across all VMs where $\forall \in$V "$\text{u}_A(\text{t})$" can be defined as "$u_i(t)$" sum of number of users getting service from v as given by Eq 2.

$$\text{S}_D(\text{t}) = (u_i(t) - u_A(t))^2 \quad (3)$$

Nonetheless, as none of the servers inform a central decision mechanism, early termination of instances is generally impossible, unless the number of requests hit the total number of running VMs.

B. Number of users

The number of users is the main decisive parameter to determine the capability of a VM instance. If the capacity of the first VM is overrun, a new VM instance is triggered. When the demand from the users tends to decrease, subsequently, the same pattern may be practiced for a cool down session. This refers to a state where all the running VM instances have less number of users when compared to their max capacity.

C. Throughput based

Most of the load balancer implementations that are based on network metrics, contrive to rely on the efficiency and adequateness of throughput, goodput, bandwidth and latency metrics [18, 25]. A decisive mechanism could trigger new instances to meet the demand by comparing the maximum carrier bandwidth, routing capability and throughput capacity of a single or a cluster of instances for the requested service by the users,.

$T_i(t) = \sum_{j=1}^{n} B_j(t)$, subject to

$$L_{Min} < T_i(t),\ L_{Max} > T_i(t) \quad (4)$$

The difference of throughput based load balancing from the other techniques is the capability to prioritize any user according to the origin of connection or application.

D. CPU or Memory capacity based

This is usually the most frequently implemented and used load balancing technique. In this technique, the requested CPU or memory load caused from the users that do not meet the total capability of the running VM instances, will trigger the instantiation of new VMs. Moreover, in order to serve more users from the same machine, there is another technique called VM migration where either the container or VM is migrated to another cloud resource that has more CPU or memory capacity availability. In order to keep the downtime at a minimum, migration must take place including all necessary memory, latest cache state. Until all this information is moved to the new machine, previous VM must continue to serve and this will keep the downtime to a minimum.

E. Hybrid Scaling Strategies

Hybrid scaling strategies are load-balancing mechanisms that are based on a collaborated understanding of application, network and cloud resource oriented objective metrics. To act as a flexible solution that can suit various circumstances, the importance of any parameter must be represented by corresponding weights. The range and the values of these weights can differ fundamentally according to the deployment strategy, corresponding usage scenarios and marketing requirements.

The constraints that introduced anticipate the concurrent availability of following items for each VM; required bandwidth, computational power and memory resources. Any of these unmet conditions might trigger a scaling activity. Cooling down in a hybrid load balancing environment shows better performance when compared to previous strategies due to the possibility of multiple termination triggers which shuts down under-utilized VMs faster.

IV. Scaling Against QoE Performance

In this section, a proposed methodology will be presented to recalibrate limited cloud resources to handle any case of QoE deterioration. The repositioning of the resources will be realized by using different load balancing techniques and a comparative resulting scheme will be provided. QoE for a user that is receiving a service from online video delivery system can be based on video player related parameters. For any HTML5 based online video player, it is easy to retrieve objective video statistics such as; initial buffering duration, number of stalls, total stall duration and resolution. There are many approaches to use these parameters and evaluate QoE for a single user [23, 25, 26]. Moreover, QoE for a cluster of users "u_v" can also be calculated that can be used as a basis to a subjective user experience. Each corresponding $Q_v(t)$

value for particular VM for υ∈V, QoE for overall system can be estimated as: $Q_v(t) = \sum_{i=1}^{u_v} \frac{Q_u(t)}{u_v}$ (5)

Conclusively, each corresponding $Q_v(t)$ value for particular VM for υ∈V, QoE for overall system can be estimated as given by: $QoE(t) = \sum_{i=1}^{u_v} \frac{Q_v(t)}{n}$ (6)

The primary benefit of a QoE based load-balancing strategy for an online video service is the prioritizing of customer satisfaction. Cooling down sessions will act in parallel to terminate active VM sessions. Unless objective video metrics across the cluster of users do not meet required minimum QoE constraints, termination of underused VMs will not take place. The following Algorithm presents a lucid understanding of the scaling triggering mechanism, which takes QoE as basis. In this methodology, each user's experience creates an impact on the overall behavior of the scaling.

ALGORITHM: QOE BASED LOAD BALANCING ALGORITHM

PREREQUISITES:

NUMBER OF USERS AT INSTANCE T FOR VIRTUAL MACHINE V; $u_v(t)$, U∈U.

0. WHILE (TRUE FOR ANY υ∈V $u_v(t)$>0)
1. MEASURE $Q_u = e^{-\frac{numStalls}{s1}} . e^{-\frac{\left(\frac{totalbuflen}{T}\right)}{s2}} . e^{-\frac{\left(\frac{bufdur}{T}\right)}{s3}}, for\ u \in U$;
2. EVALUATE Q_V $for\ \forall \in V$;
3. CALCULATE QOE FOR THE WHOLE SYSTEM $QoE(t)$.
4. CONTROL IF A SYSTEM WIDE QOE DETERIORATION IS AVAILABLE OR NOT BY CHECKING IF %50 OF THE Q_V $for\ \forall \in V$ MEET FOLLOWING CRITERIA : $Q_V < |QoE_{LIMIT}|$
5. IF (COUNT > %50 OF ∀∈V) SCALE HORIZONTALLY.
6. ENDIF.
7. FOR EACH Q_v WHERE $v \in V$
8. IF (($\Delta Q_v = Q_v(t_1) - Q_v(t_2)$) && ($\Delta Q_v < 0$) && ($|\Delta Q_v| < |S_Q|$))
9. ADD $VM\ v \in V$ TO TERMINATION QUEUE.
10. END WHILE.

V. SIMULATION ENVIRONMENT

In previous sections, an overall understanding of the load VM balancing strategies has been presented. A testbed environment has been developed to test these VM balancing strategies.

The simulation environment is built using a cluster of small-sized VM bots that consist of a light-weight Linux distribution (Ubuntu 16.04 LTS) including html5 web browsing capability (Firefox 58.0.2, Google Chrome 65 & Opera 51), which will request online content from the video service. QoE grading of each individual VM will be measured through QoE equations which are related to initial buffering time, a number of stalls, total stall duration and average resolution quality of the content [26] through the individual session. The number of these VMs will change through the testing period based on real-life data that is originated from Broadcasters' Audience Research Board (BARB) [30], providing user access statistics and rating information for a 60 minutes period. The performance of the online video platform, QoE deterioration handling approach and the cost success rate of the strategies can be compared objectively.

Figure 1 visualizes the test bed environment. The example streaming capable VM is accessible at "www.utkubulkan.co.uk/cloudqoe.html" and the corresponding QoE statistics database regarding the simulation information is publicly available through "www.utkubulkan.co.uk/cloudqoedatabase.php".

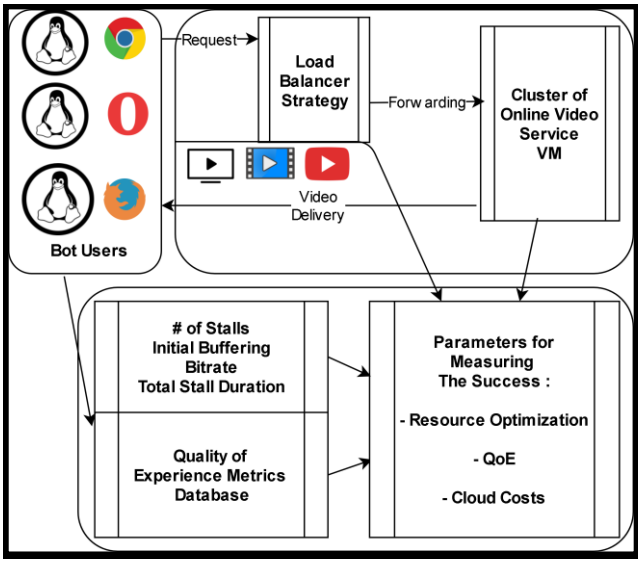

Figure 1. VM Simulation Environment

VI. COMPARISON OF LOAD BALANCING STRATEGIES

The VM load balancing testing techniques that have been introduced in the previous section are compared in the testbed environment. The results for warming up and cooling down haven been presented in terms of instantiating and terminating the VM instances. The data that have been collected and presented with cloud QoE database constitute the foundation of these inductions.

A. Warming Up Performance

The scaling strategy of a load balancer implementation has a significant impact on warming up performance and thus it can be the main bottleneck against the requested QoE levels. When the requests reach to an unexpected peak, the number of servers must scale proportionally with the demand.

Figure 2 shows the comparison of scalability strategies in terms of resource usage efficiency. The random-access implementation must be aware of the average or a total number of users that are accessing the system to be able to scale horizontally. Throughput and other resource-based strategies also show good performance especially for scenarios where the systems are optimized for prioritized user schemes. The scaling algorithms that proposed by Kesevaraja et al [23] & Chunlin et al [24] show similar performance as network oriented throughput-based algorithms, however, they lack to meet the demand of a QoE related degradation.

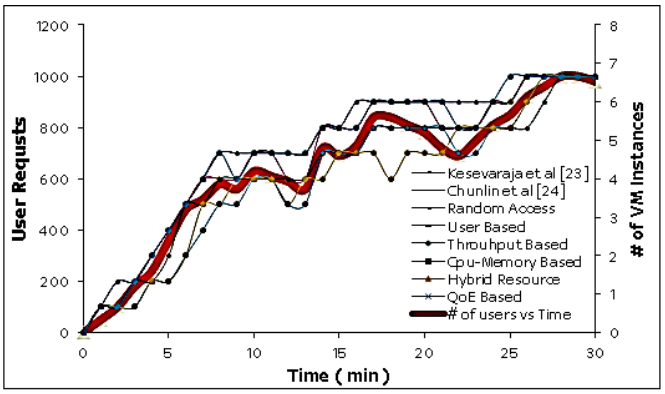

Figure 2. Resource Usage Efficiency for Different Scaling Techniques during Warming Up

B. Cooling down Performance

Cooling down strategy of an online video delivery system is as important as the warming up because this is one of the main parameters that the success rate of this implementation defines the budget estimation. In terms of cooling down, random-access shows the worst performance along with Chunlin et al [24] and QoE based scaling, as shown in Figure 3. That occurs due to the fact that average number of users that are connected to an instance can not be zero while the VMs are instantiated. So shifting the load from one server to another cannot be easily achieved. The performance of QoE based methodology guarantees customer satisfaction and prioritizes QoE, which leads to late termination of VM instances.

Due to the nature of throughput based scaling strategies, any significant drop in the throughput or minus delta between two-time epochs might be interpreted as cooling down. These instances can be marked as a low chance of selection in the priority queue for the load balancers decision mechanism. As soon as the load reaches zero where the users stop getting the service from that instance, the VM can be terminated.

In terms of costs, although Kesevaraja et al [23] & Chunlin et al [24] show good performance along with throughput and resource-based scaling strategies while cooling down, still, a conspicuous QoE degradation takes place during some of the VM termination incidents.

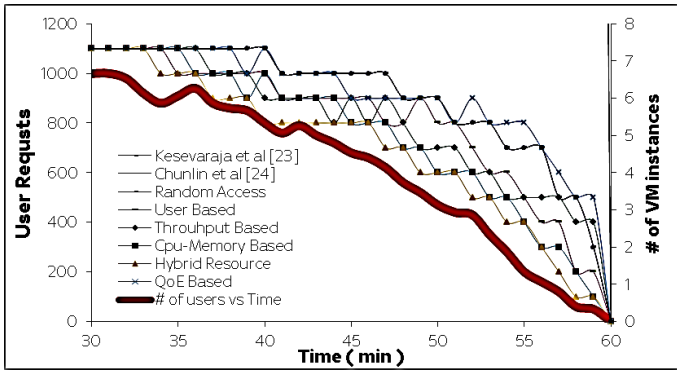

Figure 3. Resource Usage Efficiency for Different Scaling Techniques during Cooling Down

C. Scalability Strategy vs Availability

For any online video broadcasting system, availability is an important parameter. Degradation in system availability may cause increased initial buffering duration and impact expected number of stalls. Scaling strategy changes the influence of availability over QoE. Although scalability usually sounds quite flawless in many perspectives as a microservice architecture terminology, it comes with many deficiencies. One example is the transmission of the system-wide distribution of all server status, which obviously depends on the strategy, either centralized or distributed load balancer. Another one is the availability and average downtime due to new instance creation or VM migration.

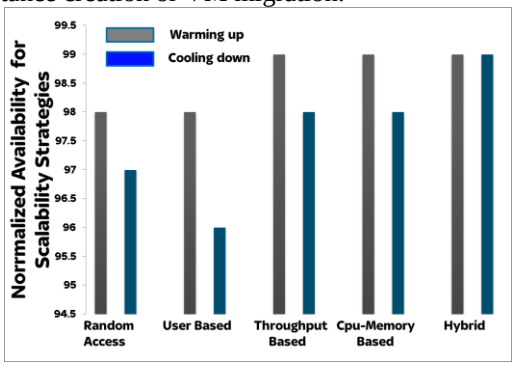

Figure 4. Availability Comparison for Scalability Strategies during Warming Up & Cooling Down

Due to its simplicity, random-access shows the best performance in terms of availability while users keep on trying new servers in the list unless a successful connection is established, as shown in Figure 4. Any new instances that are created will be added to the DNS server list. Users that request to join the service will continue to randomly try to access any of the servers. Resource-based load balancing methods show similar availability performance to the strategies where a number of users are taken as the main decision parameter.

D. Scalability Strategies vs Costs

Cloud service providers supply the needed infrastructure for the video content delivery by making available the necessary VMs instance running capability. This brings the corresponding cost for each hosted VM, where a tight delivery budget and keeping QoE for all users is a challenging task. Different scaling strategies that are provided in previous sections result in different VM costs and different budget consumption.

Due to simplistic nature of random access implementation, VM termination during cooling down is quite difficult which leads to the worst cost performance when compared with other strategies. Furthermore, CPU-memory & throughput based strategies provide acceptable warming up and cooling down strategies similarly, as shown in Figure 5. However, this may cause a tradeoff between QoE degradation and cost in some cases. The hybrid methodology offers both QoE optimization and cost maintenance. Although costs seem slightly higher than average, avoiding QoE degradation is guaranteed hence user satisfaction is considered as the primary scaling trigger.

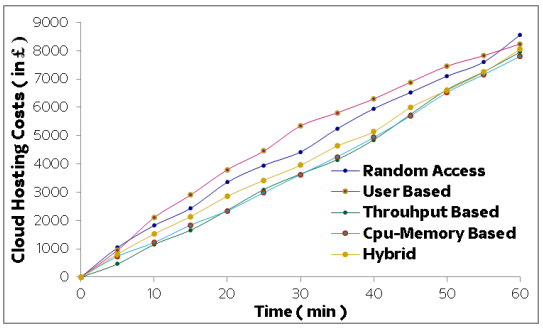

Figure 5. Cloud Hosting Costs comparison for Scaling Techniques

E. Scalability Strategies vs QoE

In terms of QoE and user satisfaction, user-based scaling methodologies shows better performance when compared to resource maintenance strategies. Especially for cases where users are not prioritized and behaved equally, scaling against users provide an acceptable performance, which is generally above average. However, for any prioritized implementation, resource-based models can provide better response to the demand in peak moments. The hybrid method introduced in this paper shows the flexibility to recover through QoE degradation and shows better performance when compared with the rest of the scaling strategies.

VII. Formulation of Computational Resources Constraints for VNF Video Streaming

In this section, formalization for memory and CPU power required to serve video streaming using VNF.

A. VNF Video Streaming

In this section, resource analysis for streaming H264 content using Apache web server will be presented. For a VNF that is responsible for video streaming, the required memory $M_{web}(t)$, computation power $C_{web}(t)$ and required storage space to operate S_{web} can be formulized as a function of bitrate, encoding type and number of users. Without loss of generality, the following arguments have been considered; a_M=175MB stands for the base memory requirement for fundamental operating system resources to operate and λ_M=0.2 is the argument representing user impact. Additionally, a_C=0.3 is the base computational usage for operating system and any additional user cause extra load on CPU with fitting arguments λ_C=0.08 and impact of encoding types are represented by $c_{encoding}$ = {266 (main profile), 133(high profile), 75 (baseline profile)}. From a general point of view, the bitrate of any video stream increases relatively to the resolution. $V_{bitrate}$ corresponds to required bandwidth for the content; 8mbit, 4mbit, 2mbit, 1mbit, 0.5mbit for resolutions 4K, 1080p, 720p, 480p, 360p accordingly.

The induction for the coefficients have been evaluated through empirical tests on a VNF hosted on Amazon Web Services (AWS) running Amazon Cloud Linux Distribution with kernel version 4.9.43-17.38.amzn1.x86_64 executing Apache/2.4.27 (Amazon) and ffmpeg 4.0.

$$M_{web}(t) = a_M + \lambda_M . e^{\frac{u_v(t).V_{bitrate}}{c_{encoding}}} \quad (7)$$

$$C_{web}(t) = a_C + \lambda_C . e^{\frac{u_v(t).V_{bitrate}}{c_{encoding}}} \quad (8)$$

$$S_{web} = \lambda_S . \frac{V_{bitrate}}{c_{encoding}} \quad (9)$$

Obviously, it is expected that any online video platform will be capable to support different bitrates and encoding types adaptive bitrate streaming. The following figure reflects the capability of a single web server against content bitrate versus number of users.

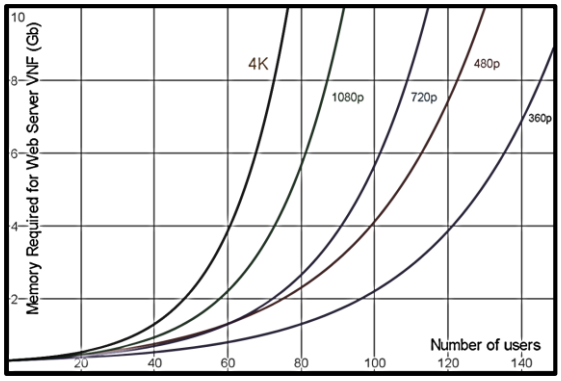

Figure 6. Memory requirement vs # of users to serve Video as a VNF

B. Transcoder as a VNF

Transcoders are the other fundamental application for any online video platform. Any uploaded mezzanine content through Content Management System (CMS) needs to be real-time encoded in order to support all connected screens at a time. The availability of transcoder VNFs shows crucial importance for the success rate of the whole delivery system. On the other hand, transcoding requires a considerably excessive amount of computational overhead. Major encoding schemes mpeg4, hevc and vp9 show different performance in terms of bitrate and storage size considering a wide range of encoding parameters.

The following figure shows the necessary amount of CPU and memory required for transcoder VM running FFMPEG on Amazon Linux where the transcoding should keep up with live streaming. The estimations correspond to physical 2.9Ghz i5 processors that are being used in AWS. Obviously, for such a task, performance degradation can be crucial and ruin QoE for the whole system.

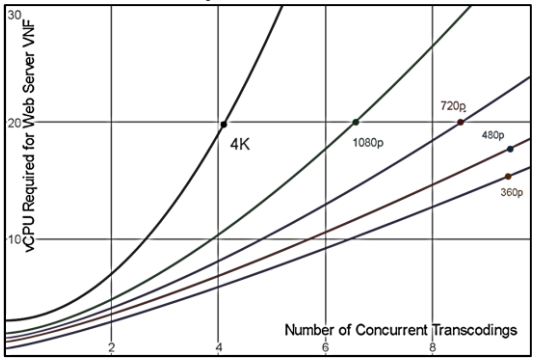

Figure 7. vCPU required for VNF vs # of Concurrent Transcoding

VIII. Conclusion and Future Work

In this work, a comparison of scaling strategies during VNF placement for online video systems have been presented using metrics such as warming up & cooling down performance, cloud hosting costs and QoE efficiency. According to the analysis, user-oriented scaling methodologies show acceptable competence on warming up durations however the cooling down efficiency lacks the adeptness to free the underused resources when compared to resource-based approaches. Throughput and computational capacity based scaling techniques show above average performance in cloud hosting costs and cooling down durations. However, they generally lack the agility to comprehend QoE degradation. To bring forward a solution for these circumstances, QoE scaling technique has been presented which considers all aspects of online video delivery that shows outstanding performance when compared with conventional cloud scaling strategies.

ACKNOWLEDGMENT

The present work was undertaken in the context of the "Self-Organization toward reduced cost and Energy per bit for future Emerging radio Technologies" with contract number 734545. The project has received research funding from the H2020-MSCA-RISE-2016 European Framework Program.

References

[1] Cisco White Paper, 2017, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf

[2] K. Chen et al, "Complexity of cloud-based transcoding platform for scalable and effective video streaming services", *Springer Science Multimedia Tools Applications, New York, 2016*.

[3] J. He et al, "Toward Optimal Deployment of Cloud-Assisted Video Distribution Services", *IEEE Transactions On Circuits And Systems For Video Technology*, Vol. 23, No. 10, March 2013.

[4] V. Stocker et al, "The growing complexity of content delivery networks: Challenges and implications for the Internet ecosystem", *Elsevier Telecommunications Policy*, Vol. 41, March 2017.

[5] K. Mokhtarian and H.-A. Jacobsen, "Flexible Caching Algorithms for Video Content Distribution Networks", *IEEE Transactions on Networking*, Vol. 25, No.2, April 2017.

[6] A K. Pathan et al, "A Taxonomy and Survey of Content Delivery Networks", *Technical Report*, [online content], http://www.cloudbus.org/reports/CDN-Taxonomy.pdf, 2007.

[7] Cisco Whitepaper, "The Cisco Content Delivery Network Solution for the Enterprise", https://www.cisco.com/c/dam/global/tr_tr/assets/docs/Enterprise.pdf, [online content], 2000.

[8] Ooyala White Paper, "How Publishers and Brands Can Build ROI with Original Video Content", http://www.ooyala.com/products/video-platform/content-management-system

[9] S. Du et al, "The Optimization of LRU algorithm based on pre-selection and cache prefetching of files in hybrid cloud", *17th International Conference on Parallel and Distributed Computing*, Applications and Technologies, Guangzhou, China, December 2016.

[10] S. Jošilo et al, "Distributed algorithms for content placement in hierarchical cache networks", *Elsevier Computer Networks*, Vol. 125, May 2017.

[11] N. Kamiyama, "Cache Replacement Based on Distance to Origin Servers", *IEEE Transactions On Network And Service Management*, Vol. 13, No. 4, August 2016.

[12] D. Pauwels et al, "A Web-Based Framework for Fast Synchronization of Live Video Players", *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, Lisbon, Portugal, July 2017

[13] X. Li et al, "Content Placement With Maximum Number of End-to-Content Paths in k-Node (Edge) Content Connected Optical Datacenter Networks", *Journal of Optical Communications and Networking*, Vol. 9, No.1, January 2017.

[14] C. Rotsos et al, "Network service orchestration standardization: A technology survey", *Elsevier Computer Standards & Interfaces*, Vol. 54, February 2017.

[15] H. He et al, "Dynamic Load Balancing Technology for Cloud-oriented CDN", *Computer Science and Information Systems*, Vol. 12, No. 2, February 2015.

[16] P. Frangoudis et al, "CDN-As-a-Service Provision Over a Telecom Operator's Cloud", *IEEE Transactions On Network And Service Management*, Vol. 14, No. 3, September 2017.

[17] C. Barba-Jimenez et al, "Cloud based Video-on-Demand service model ensuring quality of service and scalability", *Elsevier Journal of Network and Computer Applications*, Vol. 70, July 2016.

[18] S. Razzaghzadeh, "Probabilistic modeling to achieve load balancing in Expert Clouds", *Elsevier Ad Hoc Networks*, Vol. 59, May 2017.

[19] J. Yue et al, "Femto caching in video content delivery: Assignment of video clips to serve dynamic mobile users", *Elsevier Computer Communications*, Vol. 51, September 2015.

[20] C. Lin, "Strategy analysis for cloud storage reliability management based on game theory", *Journal of Computer Security*, Vol. 25, No. 2, January 2017.

[21] J. Liu, H. Sen, "A Popularity-aware Cost-effective Replication Scheme for High Data Durability in Cloud Storage", *IEEE International Conference on Big Data (Big Data)*, Washington, USA, December 2016.

[22] Y. Tang, "Achieving convergent causal consistency and high availability for cloud storage", *Elsevier Future Generation Computer Systems*, Vol. 74, September 2017.

[23] D.Kesavaraja, A. Shenbagavalli, "QoE enhancement in cloud virtual machine allocation using Eagle strategy of hybrid krill herd optimization", *Journal of Parallel and Distributed Computing*, Vol. 118, August 2018.

[24] L. Chunlin et al, "Multiple context based service scheduling for balancing cost and benefits of mobile users and cloud datacenter supplier in mobile cloud", *Elsevier Computer Networks*, Vol. 122, July 2017.

[25] K. Bilal, A. Ebrad, "Impact of Multiple Video Representations in Live Streaming: A Cost, Bandwidth, and QoE Analysis", *IEEE International Conference on Cloud Engineering*, Vancouver, BC, Canada, April 2017.

[26] ITU-T, "P.1203.3, Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport –Quality integration module", 2016.

[27] Amazon Web Services White Paper, "Cost Optimization Pillar AWS Well-Architected Framework", November 2017, [online content] https://d1.awsstatic.com/whitepapers/architecture/AWS-Cost-Optimization-Pillar.pdf

[28] L. Chen, "Supporting high-quality video streaming with SDN-based CDNs", Springer Journal of Supercomputing, USA; 2016.

[29] Microsoft, [online content] https://technet.microsoft.com/en-us/library/cc728211(v=ws.10).aspx

[30] Broadcasters' Audience Research Board, [online resource], http://www.barb.co.uk/trendspotting/analysis/online-tv-viewing/