On the Load Balancing of Edge Computing resources for on-line video delivery

Utku Bulkan, Tasos Dagiuklas, Muddesar Iqbal, Kazi Mohammed Saidul Huq, Anwer Al-Dulaimi, Jonathan Rodriguez

***Abstract*—Online video broadcasting platforms are distributed, complex, cloud oriented, scalable, micro-service based systems that are intended to provide Over-The-Top (OTT) and live content to audience in scattered geographic locations. Due to the nature of cloud VM hosting costs, the subscribers are usually served under limited resources in order to minimize delivery budget. However, operations including transcoding require high computational capacity and any disturbance in supplying requested demand might result in Quality of Experience (QoE) deterioration. For any online delivery deployment, understanding users QoE plays a crucial role for rebalancing cloud resources. In this work, a methodology for estimating Quality of Experience is provided for a scalable cloud based online video platform. The model will provide an adeptness guideline regarding limited cloud resources and relate computational capacity, memory, transcoding and throughput capability and finally latency competence of the cloud service to QoE. Scalability and efficiency of the system are optimized through reckoning sufficient number of VMs and containers to satisfy the user requests even on peak demand durations with minimum number of VMs. Both horizontal and vertical scaling strategies (including VM migration) are modelled to cover up availability and reliability of intermediate and edge Content Delivery Network (CDN) cache nodes.**

***Index Terms*—QoE, Cloud, Virtual Machines, Dockers, Scalability, Availability, Reliability, Mathematical Modelling, Online Video Platform, Content Management Systems.**

I. Introduction

Online video market has been growing exponentially for the last decade. Globally, IP video traffic will be 82 percent of all consumer Internet traffic by 2021 [1]. Internet video will continue to grow at a rapid pace, increasing 3.6-fold by 2021. Each day, users request for more content and new services are being launched to confront the growing demand. More 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 has evolved to keep the requested content cached in the nodes that are closer to the users. This has leaded to breakthrough in efficiency by many aspects, including service capacity, reduced latency and better cache management [6]. The procedure starts with the first request from the user and the caching follows a pull model [7] unless a pre-push model [8] is either configured or scheduled via Content Management System (CMS) [9]. The requests trigger the intermediate and edge nodes to copy the content which sorts it in a better accessible state for the users. The content that is frequently used and accessed stay in the cache longest time in alignment with a special purpose priority queue [10]. Depending on different CDN deployments, the distributed cache nodes may have the capability to search other nodes caches [11] for a requested content and copy it from a closer and cost efficient neighboring node. Current academic research viewpoint [5,7] and state of art technology point of view [3, 8, 9] provide an understanding that only relies on objective network metrics and cloud resource constraints where this paper introduces a brand new foundational understanding of the impact of QoE on load balancing and resource optimization.

Fig. 1 presents an overall visual of the propagation of origin content throughout a CDN. Following the triggering action of content being requested by the user, the origin [12] copies (before copying, an on the fly transcode might take place in order to support other resolutions or codec profiles) the content to the intermediate CDN node. According the size of the deployment of the requested content catalogue the number of intermediate cache layers might vary. Conclusively, the edge CDN cache pulls the content, and end-users get the service via their video players [13].

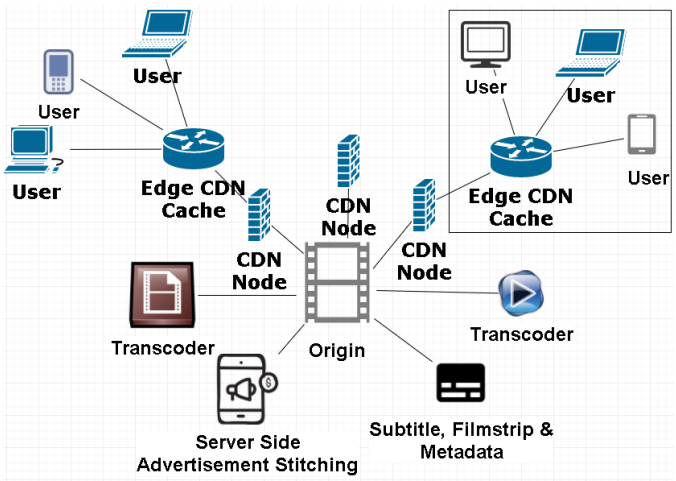

Fig. 1. Diagram for scalable online video delivery platform

U. Bulkun, T. Dagiuklas and M. Iqbal are with London South Bank University, UK. E-mail:{bulkanu; tdagiuklas; m.iqbal}@lsbu.ac.uk

K. M. S. Huq, and J. Rodriguez are with the Instituto de Telecomunicações, Aveiro, Portugal. E-mail:{kazi.saidul; smumtaz; jonathan}@av.it.pt A. Al-Dulaimi is with EXFO Inc. Canada E-mail: anwer.aldulaimi@utoronto.ca

TABLE I
LIST OF NOTATIONS

Notation	Meaning
$u_A(t)$	Average number of users across all VMs where $\forall v \in \mathcal{V}$.
$u_i(t)$	Number of users getting service from VM v, $\forall v \in \mathcal{V}$.
$S_D(t)$	Standard deviation for number of user vs average.
S	Average abnormality in number of users for each VM v.
$u_L(t)$	Lower limit for # of users for a VM v,$\forall v \in \mathcal{V}$ to be added into a termination priority queue.
$B_i(t)$	Bandwidth requested by a single user $\forall U \in \mathcal{V}$.
$T_i(t)$	Total throughput used by the VM v, $\forall v \in \mathcal{V}$.
L_{Min}	Minimum throughput limit for a VM v, to exist.
L_{Max}	Maximum throughput limit for a VM v to operate and the limit to trigger system to scale up.
$T_{pi}(t)$	Total prioritized throughput for different ranked users getting service from a VM.
$p(t)$	Prioritization parameter for a user.
$C_M(t)$	Total computational power used by a VM v, $\forall v \in \mathcal{V}$.
$C_i(t)$	Computational power required from VM by a single user u, $\forall U \in \mathcal{U}$.
$M_M(t)$	Total memory used by a VM v,$\forall v \in \mathcal{V}$.
$M_i(t)$	Memory required from VM by a single user u, $\forall U \in \mathcal{U}$.
L_c	Maximum CPU power limit for a VM v to operate and the limit to trigger system to scale up.
L_M	Maximum memory usage limit for a VM v to operate and the limit to trigger system to scale up.
S_c	Scalability parameter
μ_c	Ratio for system wide Cpu capacity vs System wide Cpu limit to trigger scaling.
μ_m	Ratio for system wide memory capacity vs System wide Cpu limit to trigger scaling.
μ_{LC}	Ratio for single VM $\forall v \in \mathcal{V}$ Cpu capacity vs Cpu limit to trigger scaling.
μ_{LM}	Ratio for single VM $\forall v \in \mathcal{V}$ memory vs memory limit to trigger scaling.
μ_v	Ratio for current # of users vs users limit for each VM.
D	Evaluated vector for hybrid decision approximation on scaling.
d_i	Components of the vector D
α_i	Weights for each hybrid decision parameter of vector A
$Q_u(t)$	Quality of Experience (QoE) for single user
numstalls	Number of stalls during in a watch session for a user
totalbuflen	Total buffer duration during in a watch session
bufdur	Initial buffering duration in a watch session.
s1, s2, s3	Weights for objective video metrics for QoE.
$Q_v(t)$	QoE for a VM v, $\forall v \in \mathcal{V}$
QoE(t)	System wide QoE
QoE_{LIMIT}	The QoE limit for triggering a scaling incident
ΔQ_v	Delta of VM QoE between two Q instances.
$M_{web}, C_{web}, S_{web}$	Total amount of memory and cpu power and storage that is required for a web server to operate.
a_M, a_C, a_S	Base required memory and cpu power for a web server to execute
$\lambda_M \lambda_C$	Weight representing the impact of memory and cpu power for each user taking service from web server.
$V_{bitrate}$	Bitrate required for a single video stream
$c_{encoding}$	Constant corresponding encoding type for h264.
$u_{Trans}(t)$	Number of concurrent transcoding activities running in a VNF

The architecture of the working mechanism of edge content nodes [14] involves cache content copy that resides in a VM that is pulled from origin and containers that mount to this VM via Network File System (NFS) [15]. Actual contact points for the users are these front line load balancers [16] that redirect requests to the containers [17], which deliver the chunks of video data to the players executing on user consumer devices. The number of web servers must vary in time due to the variations in number of users that try to access the service. Correspondingly, any of these VMs running in the system increases the cloud associated costs. Therefore, optimizing the number of running instances [18] in the cloud plays a crucial role in lowering the cloud hosting costs.

Considering the procuring of broadcasting rights [2] for major events (such as Super Bowl or Eurovision) which require a huge budget, livestreaming companies try to avoid any additional costs whenever possible [13]. On the other hand, any unexpected peak in user requests result in a parallel unforeseen scalability demand and equivalent un-predicted costs on cloud. An attempt to confront this demand by redundancy requires other additional investment on redundancy [3] which is also usually neglected. On the other hand, the whole system might still not conclude as an error prone service in terms of user satisfaction, considering the main foundation of the scalability might not rely on QoE but other indicators such as network metrics or resource restrictions only [5], [7], [9].

To overcome the limitations that has been addressed, the primary intention of this paper is to cover the demands of state of art scalable online video delivery systems by comparing and presenting different load balancing strategies [19] to provide a guidance for rebalance the limited cloud resources and maintain QoE while keeping delivery budget constraints [20] in consideration. Finally, a set of equations will be presented which relates video metrics with cloud resources including cpu power, memory and throughput.

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

II. Related Works

Defining a scalable methodology for cloud based services has attracted a lot attention due to the demand for distributed applications that provide reliability [21], duration [22] and availability [23]. In their recent work, Kesevaraja et al has modelled [24] single VM instance as Eq. 1, where γ_{PN} is the success rate of a physical node in percentage, $R_{Processor}$ current utilization of the processor in GHZ, $C_{Processor}$ the maximum capacity of the processor in GHZ, R_{Memory} current utilization of primary memory in MB, C_{Memory} the maximum available capacity of primary memory in MB, $R_{Databits}$ the data bits transferred among time interval $T_{Interval}$, $C_{Bandwidth}$ the bandwidth of the network in bits. Although, this model provides a good understanding of the single local node, still, it lacks the impact of video metrics related parameters like number of stalls or initial buffering duration as it is formulized with Eq. 13 & Eq. 14.

$$\gamma_{PN} = K\Big(\frac{R_{Processor}}{C_{Processor}} + \frac{R_{Memory}}{C_{Memory}} + \frac{R_{Databits}}{C_{Bandwidth} * T_{Interval}}\Big) \tag{1}$$

Chunlin et al has proposed multiple context based service scheduling model [24] that adopts network utility maximization framework to maximize total system utility. When the mobile device applications job is accepted by the cloud system, it is scheduled and assigned to the cloud resource according to the system context. The utility function U for multiple context based service scheduling optimization is denoted in Eq. 2, where r_{ij} refers to mobile users unit cost and q_{ij} requirement of mobile device for storage, CPU, RAM and bandwidth respectively. Although this proposition offers a good understanding of cloud resource efficiency, nevertheless it does not consider the impact of QoE. Hence, the competence of the model cannot fulfil the demands of a state of art online video platform where QoE based load balancing Algorithm IV shows superior performance.

$$U = \sum_{i=1}^{I} \big(r_{ij}^{storage} \log x_{ij}^{storage} + r_{ij}^{cpu} \log x_{ij}^{cpu} + r_{ij}^{ram} \log x_{ij}^{ram} + r_{ij}^{bandwidth} \log x_{ij}^{bandwidth}\big) \tag{2}$$

Bilal et al has provided a formula [25] for cloud costs with Eq 3, where $Cost_t$ is refered on computational instances, c_t is Amazon c4.large instance price, second double sum D is total amount of data in bits per second required for server v viewers watching a specific channel, $Cost_d$ is total cost for D data per second. This formulation gives an illustration of cost, bandwidth and QoE analysis for delivering online video; yet, it provides a picture of single video server and lacks the representation of scalability through microservices and distributed systems.

$$Cost = \sum_{r \varepsilon R} c_t + cost_d . \sum_{r \varepsilon R} \sum_{v \varepsilon V} d_r . I_{v \varepsilon V} \tag{3}$$

In ITU-T P.1203.3 recommendation [26], a media session quality score is formulated based on number of stalls, total stall duration, buffering duration. This provides a basis for a single users watching experience in terms of simple video player metrics.

$$OOE = e^{-\frac{numStalls}{s1}} . e^{-\frac{(\frac{totalbuflen}{T})}{s2}} . e^{-\frac{(\frac{bufdur}{T})}{s3}} \tag{4}$$

As opposing the works that are available in academic literature [20], [22]–[25], this paper provides a hybrid scalability model that considers cost, resource efficiency and more importantly QoE aspects of online video delivery through cloud computing. Comparison of pros & cons for different scaling strategies is presented to cover up various scenarios. Additionally, formulization 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 Mobile Edge Computing

For solving a limited resource vs cluster of user problem, load balancers usually provide the most efficient solutions. 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 clarified.

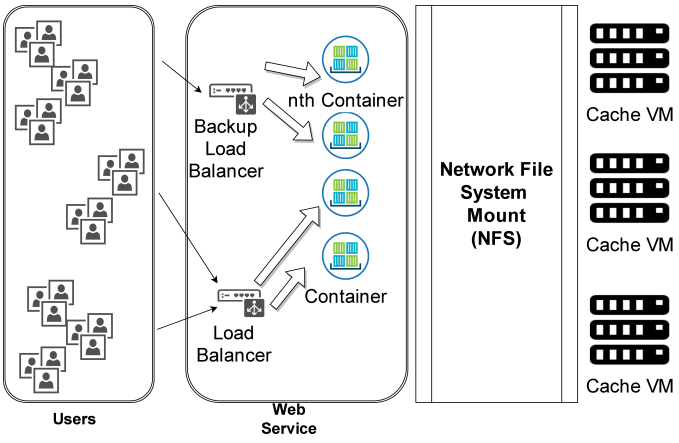

Fig. 2. Edge CDN Cache Architecture (This figure represents right-top square of Fig 1)

A. Random-access (aka Round-Robin)

Random-access load balancing technique works on the assumption that the users should connect randomly to any server through a list of available servers.

As depicted with Fig. 3, statistically (with the increasing total number of users) all the nodes will congregate to have equal number of users [28]. In this case, the definition of randomness or the range of random generation capability becomes an important fact and strictly related to the expected number of users that intent to use the service.

$$u_A(t) = \frac{\sum_{i=0}^{v} u_i(t)}{v} \tag{5}$$

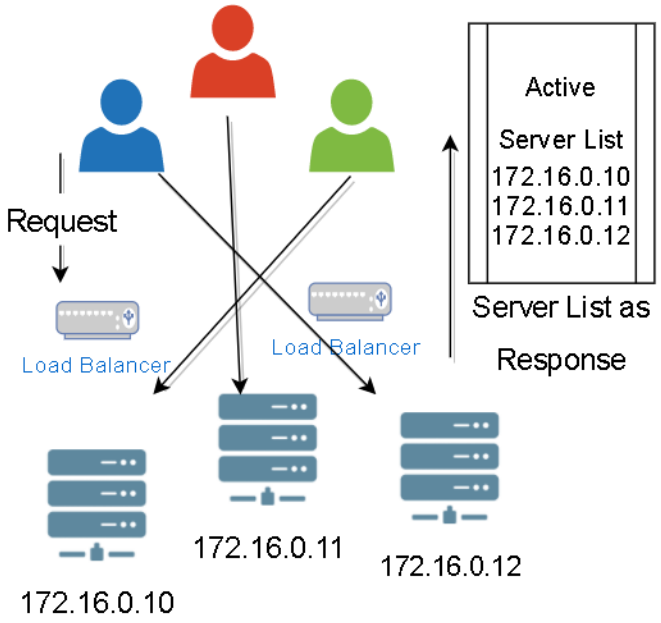

Fig. 3. Random-Access Load Balancing Strategy

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

$$S_D(t) = (u_i(t) - u_A(t))^2 \tag{6}$$

Standard deviation of the average number of users is given by Eq. 3 and calculated from each VM separately statistically converges to zero due to the random distribution of the users during "warming up" session.

$$S^2 = \sum \frac{(u_i(t) - u_A(t))^2}{i-1} \tag{7}$$

In case of growth of either the number of users or number of VMs will result in this convergence of sum of standard deviation of load to zero as given by Eq. 4 & Eq. 5.

$$\lim_{v \to \infty} s^2 = 0 \tag{8}$$

Cooling down procedure for random access load balancing is reasonably straightforward. Nonetheless, as none of the servers are informing a central decision mechanism where all the information regarding the server capacity statistics are stored and analyzed, early termination of instances is generally impossible to establish unless the number of requests hit the total number of running VMs.

B. Number of users

For this load balancing technique, 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, following this activity, a new VM instance is run to meet the demand. 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 to have less number of users when compared to their max capacity. New requests will be handled by the already running instances that are in the highest rank in the queue, so the demand can be met by less number of VMs and the remaining VMs can be terminated when they are not serving any more users. Obviously, relying only on number of users will also result in treating each user equally and not having capability to classify users as premium or with prioritized QoE levels.

Algorithm 1: Load Balancing Algorithm Based on The Number of Users

PREREQUISITES:
NUMBER OF USERS AT INSTANCE T FOR VIRTUAL MACHINE V; $u_v(t), U \in \mathcal{U}$.
0.WHILE (TRUE FOR ANY $v \in \mathcal{V}$ $u_v(t) > 0$)
1.CALCULATE EQ. 2, $S_D(t) = (u_i(t) - u_A(t))^2$ FOR EACH $v \in \mathcal{V}$,
2. FIND $MAX_N S_D(t)$, GET SERVERID N.
3. ROUTE USER $U \in \mathcal{U}$ TO NTH SERVER.
4. FOR EACH $v \in \mathcal{V}, u_v(t) \leq u_L(t)$ ADD V TO THE COOLING DOWN QUEUE, WHERE $u_L(t)$ REFERS TO THE LOWER LIMIT FOR # OF USERS FOR A VM V, $\forall v \in \mathcal{V}$ TO BE ADDED INTO A TERMINATION PRIORITY QUEUE.
5. ROUTE NEW USERS TO THE SERVERS EXCLUDING $u_v(t)$ AND THE REST OF THE TERMINATION QUEUE.
6. CHECK IF $u_v(t) = 0$, TERMINATE $u_v(t)$
8. ENDWHILE

C. Throughput Based

Most of the load balancer implementations that are based on network metrics contrive relying on the efficiency and adequateness of throughput, goodput, bandwidth and latency [18, 25].

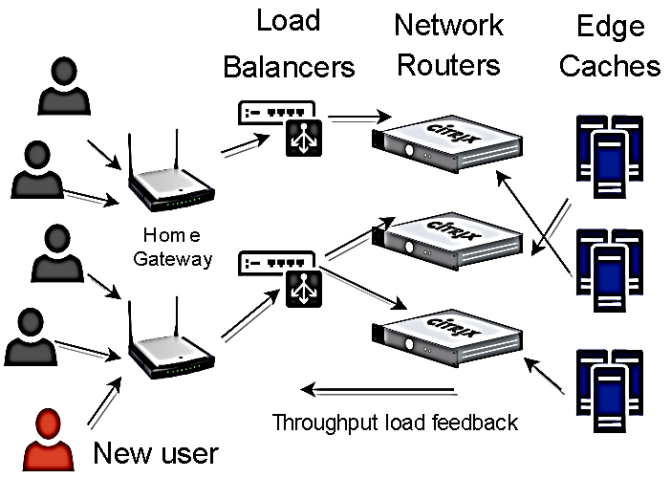

Fig. 4. Throughput Based Load Balancing Strategy

Algorithm 2: Throughput Based Load Balancing Algorithm

PREREQUISITES:
NUMBER OF USERS AT INSTANCE T FOR VIRTUAL MACHINE V; $u_v(t), U \in \mathcal{U}$.
CPU AND MEMORY LOAD ON INSTANCE $v \in \mathcal{V}; C_i, M_i$, CPU AND MEMORY LIMIT FOR $u_v; L_C, L_M$, SCALABILITY PARAMETER S_C.
0. WHILE (TRUE FOR ANY $v \in \mathcal{V}$ $u_v(t) > 0$)
1. ESTIMATE $\mu = \sum_{i=1}^{n} C_i(t)/(n, L_c); \forall v \in \mathcal{V}$
2. FOR $\forall v \in \mathcal{V}$, IF $(C_i(t) > L_C$ && $M_i > L_M) COUNT++$;
3. IF $(COUNT > S_C)$ SCALE HORIZONTALLY.
4. ENDIF
5. ENDFOR
6. END WHILE

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, a decisive mechanism could trigger new instances to meet the demand.

$$T_i(t) = \sum_{j=1}^{n} B_j(t) \tag{9}$$

$$\begin{aligned} L_{Min} &\leq T_i(t) \\ L_{Max} &\geq T_i(t) \end{aligned} \tag{10}$$

The difference of throughput based load balancing from the other techniques is the easy capability to prioritize any user according to the origin of connection or application which uses a particular prioritization API. The prioritization factor that is represented in Eq. (7) refers to QoS parameter which refers to the service type.

TABLE II
NETWORK THROUGHPUT QOS PRIORITIZATION PARAMETER TABLE

Service Type	Prioritization Factor
Standard User	0.7
Silver User	0.75
Gold User	0.80
Premium User	0.87
Guaranteed Service	0.99

$$T_{pi}(t) = \sum_{j=1}^{n} p(t) B_j(t) \tag{11}$$

Warming up and cooling down sessions may rely on the demand of the highlighted throughput based on the content request which results in a relatively easier judgment for a cooling down practice when compared to random-access based load balancing methods.

D. CPU or Memory capacity based

This is usually most frequently implemented and used load balancing technique, where the requested CPU or memory load caused from the users do not meet the total capability of the running VM instances, which will be met by instantiating new VMs. Moreover, in order to serve more users from the same machine, there is another technique called VM migration where the container or VM is migrated to another cloud resource that has more cpu or memory capacity available.

$$C_M(t) = \sum_{i=0}^{n} C_i(t) \tag{12}$$

$$M_M(t) = \sum_{i=0}^{n} M_i(t) \tag{13}$$

In order to keep the down time at minimum, migration must take place including all necessary memory, latest cache state. Until all this information is moved to the new machine, previous VM should continue to serve and this will keep the down time at minimum. Beneficial side of VM migration is to keep legacy systems working without a modern load balancing technique. However, it is obvious that most of the online video platforms are micro-service based and VM migration would not suit a geographically distributed content delivery.

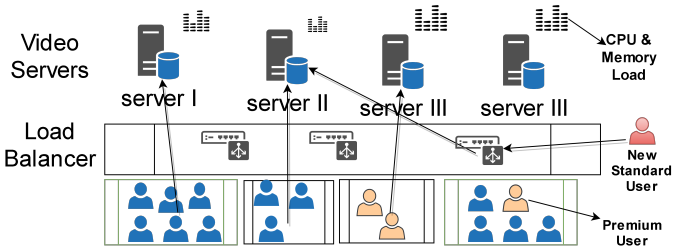

Fig. 5. Physical Resource (CPU and Memory) Based Load Balancing Strategy

E. Hybrid Scaling Strategies

Hybrid scaling strategies are load balancing mechanisms that are based on a collaborated understanding of network and cloud resource oriented objective metrics. To act as a flexible solution that can suit to various circumstances, the importance of any parameter will 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.

$$\begin{aligned} B(T) &< \sum_{i=1}^{n} T(t) \\ L_c &< \sum_{i=0}^{n} C_i(t) \\ L_M &< \sum_{i=0}^{n} M_i(t) \\ u_v(t) &\leq u_L(t) \end{aligned} \tag{14}$$

The constraints that are given with Eq. (11) anticipates the concurrent availability of following items; required bandwidth should be correlated and satisfy the required throughput for each VM, computational power and memory resources should

be more than the requested limits LC & LM and furthermore, number of users assigned for each VM should be less than the limit serving capability of a VM. Any of these unmet conditions might trigger a scaling activity.

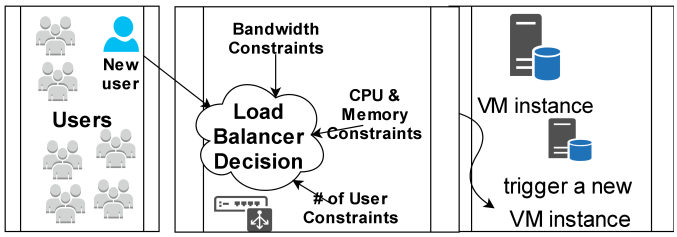

Fig. 6. Hybrid Load Balancing Strategy

Fig. 6 visualizes the hybrid scaling strategy where the impact of the constraints might trigger a new VM instance. In Eq. 1, Kesevaraja et al [23] has formulized the picture in a similar manner. Cooling down in a hybrid load balancing environment shows comparatively better performance when compared to previous strategies due to the possibility of multiple termination triggers which shuts down underused or unused VMs faster.

Algorithm 3: Hybrid Load Balancing Algorithm

PREREQUISITES:
NUMBER OF USERS AT INSTANCE T FOR VIRTUAL MACHINE V; $u_v(t), U \in \mathcal{U}$.
LIMIT NUMBER OF USERS FOR A VM AT INSTANCE $T; u_L(t)$. CPU AND MEMORY LOAD ON INSTANCE $v \in \mathcal{V}; C_i, M_i$, CPU AND MEMORY LIMIT FOR $u_v; L_C, L_M$, SCALABILITY PARAMETER S_C.
0. WHILE (TRUE FOR ANY $v \in \mathcal{V} u_v(t) > 0$)
1. ESTIMATE $\mu c = \sum_{i=1}^{n} C_i(t)/(n, L_c); \forall v \in \mathcal{V}$;
2. ESTIMATE $\mu M = \sum_{i=1}^{n} M_i(t)/(n, L_M); \forall v \in \mathcal{V}$;
3. FOR $\forall v \in \mathcal{V}, \mu Lc = C_i(t)/Lc\ \mu LM = M_i/LM$;
4. $\forall v \in \mathcal{V} \mu v = u_v(t)/u_L(t)$;
5. COMPARE α_i TO d_i COMPONENTS OF THE DECISION VECTOR D.
6. IF $(|\alpha_i - d_i| > S_{Ci})$ THEN SCALE HORIZONTALLY.
7. ELSE FIND MIN D_I, ADD IN PRIORITY QUEUE FOR TERMINATION.
8. ENDIF
9. ENDFOR
10. END WHILE

Algorithm 3 provides the step by step operational procedure for hybrid load balancing technique. Estimation of the system parameters against constraints and comparing the current state of the system resources to components of the decision vector forms the foundation of the decision mechanism for this procedure. If the VM meets the underused condition then it will be queued for termination.

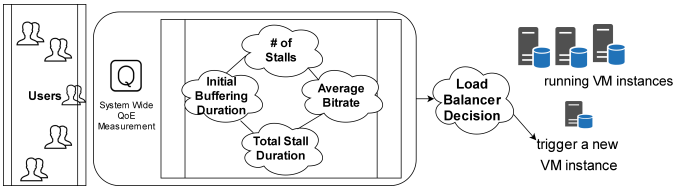

Fig. 7. QoE oriented Load Balancing Strategy

IV. Scaling Against QoE Performance

In this section, a methodology will be provided 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 with regards to QoE. QoE for a user that is receiving a service from online video delivery system can be based on video player related parameters. These parameters correspond to the subjective feeling of continuity and subsequently uninterrupted watch. For any HTML55 or mobile app 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 oriented scalability strategy as given with Eq. 15. Conclusively, each corresponding $Q_v(t)$ value for particular VM for $\forall v \in \mathcal{V}$ QoE for overall system can be estimated.

$$Q_v(t) = \sum_{i=1}^{u_v} \frac{Q_u(t)}{u_v} \tag{15}$$

Conclusively, each corresponding $Q_v(t)$ value for particular VM for $\forall v \in \mathcal{V}$ QoE for overall system can be estimated as given by Eq. 16.

$$QoE(t) = \sum_{i=1}^{u_v} \frac{Q_v(t)}{n} \tag{16}$$

Algorithm 4 gives a lucid understanding of the scaling triggering mechanism which takes QoE as basis. In this methodology, each users experience creates an impact on the overall behavior of the scaling attitude.

The primary benefit of a QoE based load balancing strategy for an online video service is the attitude of prioritizing customer satisfaction. Any degraded customer satisfaction across a cluster of subscribers will trigger a scaling incident which will result in optimized QoE under any circumstance.

Cooling down sessions will act in parallel; the mechanism responsible for the termination of active VM sessions will still keep QoE in consideration of primary importance. Unless objective video metrics across the cluster of users do not meet required minimum QoE constraints, termination of underused VMs will not take place.

Algorithm 4: QOE Based Load Balancing Algorithm

PREREQUISITES:
NUMBER OF USERS AT INSTANCE T FOR VIRTUAL MACHINE V; $u_v(t), U \in \mathcal{U}$.
0. WHILE (TRUE FOR ANY $v \in \mathcal{V} u_v(t) > 0$)
1. MEASURE $Q_u = e^{-\frac{numStalls}{s1}}.e^{-\frac{(\frac{totalbuflen}{T})}{s2}}.e^{-\frac{(\frac{bufdur}{T})}{s3}}$, for $\forall u \in \mathcal{U}$.
2. EVALUATE Q_V for $\forall v \in \mathcal{V}$;
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 $\forall v \in \mathcal{V}$ MEET FOLLOWING CRITERIA : $Q_V < |QoE_{LIMIT}|$
5. IF (COUNT $>$ %50 OF $\forall v \in \mathcal{V}$ SCALE HORIZONTALLY.
6. ENDIF
7. FOR EACH Q_v WHERE $\forall v \in \mathcal{V}$
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 $\forall v \in \mathcal{V}$ TO TERMINATION QUEUE.
10.END WHILE

A. Details of the Simulation

In previous sections, an overall understanding of the load balancing strategies and their particular performance details for various circumstances has been presented. Subsequently, to test these methodologies in a controlled test bed environment, a simulation technique is going to be proposed and the details of the simulation will be clarified.

The simulation technique 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 bot will be measured through QoE equations which are related to initial buffering time, number of stalls, total stall duration and average resolution quality of the content [26] through individual session. The number of these bots will change through the testing period based on real-life data that is originated from Broadcasters Audience Research Board (BARB) [30] which provides user access statistics and rating information for a 60 minutes period. According to the performance of the load balancers and harmoniously, the performance of the online video platform, QoE deterioration handling approach and the cost success rate of the strategies can be compared objectively.

Fig. 8 visualizes the test bed environment, the relationship of bot users, load balancer, VMs responsible for streaming and QoE database. 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 via username and password publicbot.

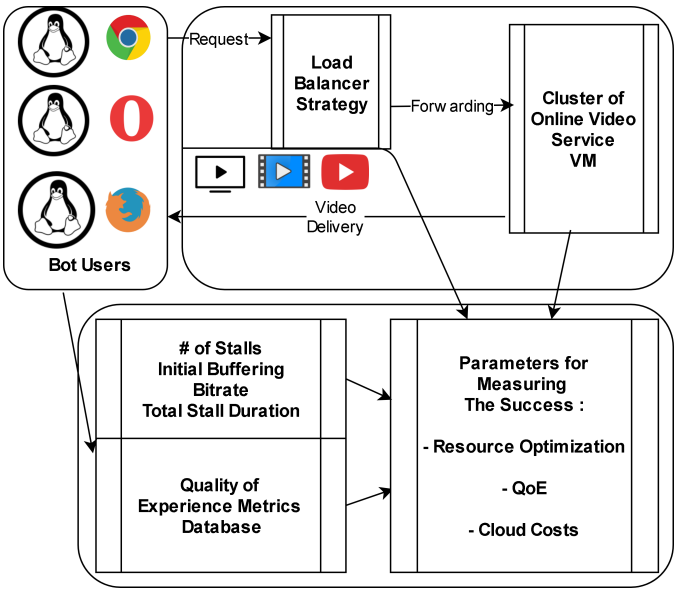

Fig. 8. Details of the Simulation Method consisting of Bot Users and the introduction of measuring the success via parameters of resource optimization, QoE and cloud costs

V. Comparison of Load Balancing Strategies and Discussions of Scaling Performance

The bot based load balancing testing technique that has been introduced in the previous section has been executed for each scaling strategy including Kesevaraja et al, Chunlin et al [24], random access, user based, throughput based, cpu-memory based, hybrid resource and QoE based. The results for warming up and cooling down has been separately presented due to the foundational differences regarding the demand of instantiating and terminating the VM instances. The data that has been collected and presented with cloud QoE database constitutes the foundation of these inductions and figures 9 & 10.

A. Warming Up Performance

The scaling strategy of a load balancer implementation has a significant impact on warming up performance where reliability of a new VM instance can be the main bottleneck against the requested QoE levels. When the requests reach to an unexpected peak on a geography which does not have the required content already cached, to serve the users within expected time frame, the number of servers should scale proportionally in correspondence with the demand. If this request is not met, overall response quality is not acceptable from a deployment point of view.

The Fig. 9 shows the comparison of scalability strategies shows that scalability based on random-access and number of users load balancing shows good warming up performance against QoE. However, random-access implementation must be aware of the average or total number of users that are accessing the system to be able to scale horizontally. Throughput and other resource based strategies also shows 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] shows similar performance as network oriented throughput based algorithms, however they lack to meet the demand of a QoE related degradation.

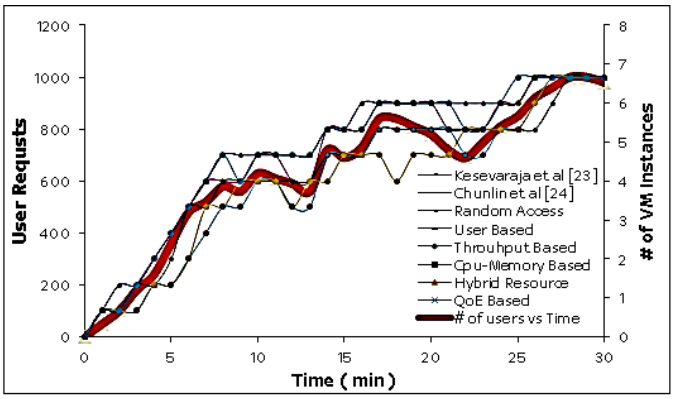

Fig. 9. Resource Usage Efficiency for Different Scaling Techniques during Warming Up

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, The bottleneck for random-access for this metric results due to the circumstance when any of the VMs are instantiated, the average number of users that are connected to any instance cannot be zero (unless all instances have zero connections), and without having any other indicator, the users are going to continue to connect to all random servers. So shifting the load from one server to another would not be easily achieved.

The performance of QoE based methodology guarantees customer satisfaction and prioritize QoE which leads to late termination of VM instances. Although this results in higher cloud costs, the impact of customer happiness can be regarded as future investment and long term customer engagement.

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, and these instances can easily be marked as low chance of selection in priority queue for the load balancers decision mechanism and as soon as the load reaches zero where the users totally stop getting the service from that instance, the VM can be terminated. Also resource based strategies shows good performance on cooling down cases.

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

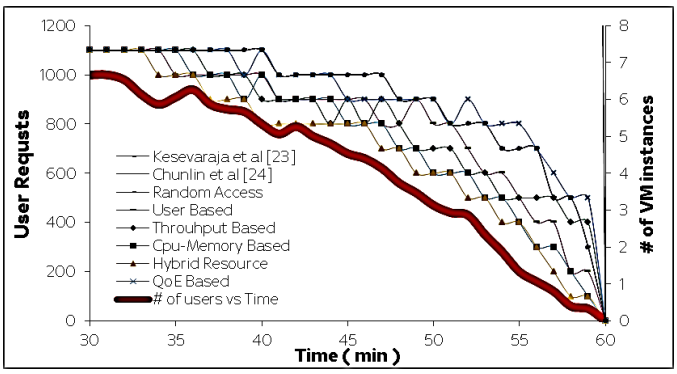

Fig. 10. Resource Usage Efficiency for Different Scaling Techniques during Cooling Down

C. Scalability Strategy vs Availability

For any online video broadcasting system, availability is a critically important concept. Degradation in system wide average availability may cause increased initial buffering duration and impact expected number of stalls which will cause QoE deterioration. Scaling strategy changes the influence of availability over QoE.

Although scalability usually sounds quite flawless in many perspectives as a micro service 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.

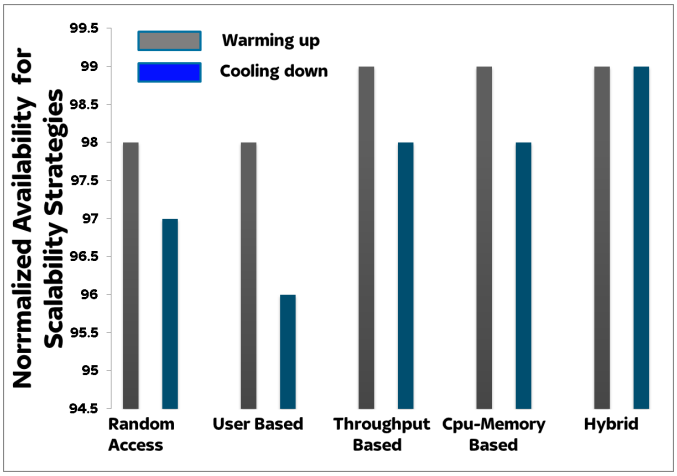

Fig. 11. Availability Comparison for Scalability Strategies during Warming Up & Cooling Down

Due to its simplistic nature random-access shows the best performance in terms of availability, where users keep on trying new servers in the list unless a successful connection is established. Any new instances that are created will be added to the simple DNS like server list, and users that requests to join the service, will continue to randomly try to access any of the servers in the list. Resource based load balancing methods will also show similar availability performance to

the strategies where number of users are taken as the main decision parameter. Statistics for the downtime of a cache or webserver VM instance generally provides the average availability level for the system.

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. Obviously, this brings the corresponding cost for each hosted VM. Proceeding with a tight delivery budget and keeping QoE for all customers in expected levels can be a challenging task. Different scaling strategies that are provided in previous sections result in different VM costs and different budget consumption.

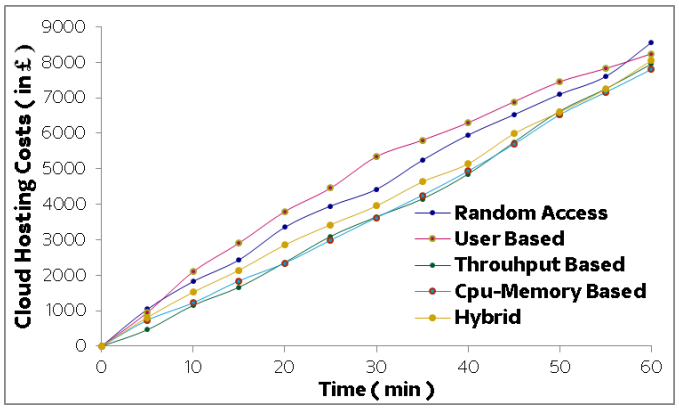

Fig. 12. Cloud Hosting Costs comparison for Scaling Techniques

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. Following that cpu-memory & throughput based strategies provide acceptable warming up and cooling down strategies concurrently. Still, this may cause a tradeoff between QoE degradation and cost on some cases. Hybrid methodology that is provided in this paper offers both QoE optimization and cost maintenance. Although costs seems lightly higher than average, avoiding QoE degradation is guaranteed hence user satisfaction is considered as a main scaling indicator.

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 like throughput or cpu & memory. 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 scale better providing 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.

VI. Formulation of Computational Resources Constraints for Online Video Streaming via VNFS

In this section, a formulization for memory and cpu power required to serve video through a VNF that will operate in an online video platform will be presented.

A. Web Server as a VNF

From a general point of view, the bitrate of any video stream increases when the resolution increases considering the encoding type same. For streaming a main profile h264 video content using apache web server, required memory $M_{web}(t)$, computation power $C_{web}(t)$ and required storage space to operate S_{web} can be formulized with Eq 17, 18 & 19 as a function of bitrate, encoding type and number of users where the arguments have following values: $a_M = 175MB$ and $\lambda_M = 0.2, a_C = 0.3, \lambda_C = 0.08, c_{encoding}$ = 266 (main profile), 133(high profile), 75 (baseline profile). $V_{bitrate}$ corresponds to 8mbit, 4mbit, 2mbit, 1mbit, 0.5 for 4K, 1080p, 720p, 480p, 360p accordingly. The induction has been evaluated using Amazon Cloud Linux Distribution with kernel version 4.9.43-17.38.amzn1.x86_64 running Apache/2.4.27 (Amazon).

$$M_{web}(t) = a_M + \lambda_M . e^{\frac{u_v(t).V_{bitrate}}{c_{encoding}}} \tag{17}$$

$$C_{web}(t) = a_C + \lambda_C . e^{\frac{u_v(t).V_{bitrate}}{c_{encoding}}} \tag{18}$$

$$S_{web} = \lambda_S . \frac{V_{bitrate}}{c_{encoding}} \tag{19}$$

Obviously, it is easily expected that any online video platform will be capable of state of art adaptive bitrate streaming while supporting different bitrates and encoding types. Table III represents a standard user scheme where each user is behaved as equal and expected to watch a conventionally standard encoded content. Fig. 13 reflects the capability of a single web server against content bitrate (which is related to content resolution) and number of users.

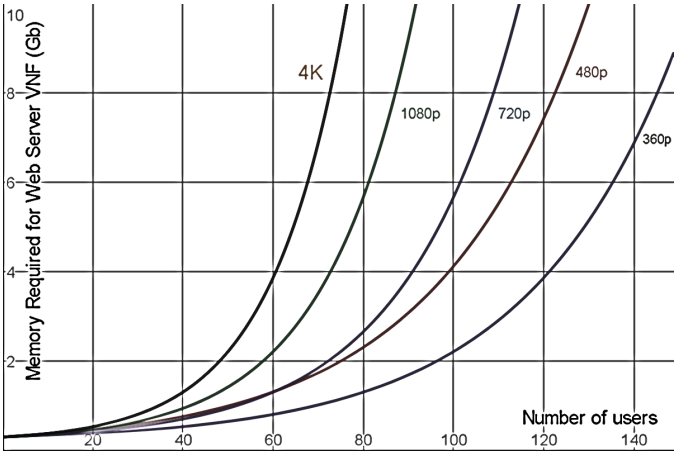

Fig. 13. Memory requirement vs Number of users to serve Online Video for varying resolutions (bitrates) via Web Server as a VNF

TABLE III
CLOUD INSTANCE RESOURCE COMPARISON AGAINST MAX SUPPORTED # OF USERS FOR STREAMING MAIN PROFILE H264 4MB 1080P CONTENT

Amazon Cloud Vm Standard Instance Flavors	VCPU	Memory (in GB)	Dedicated Bandwidth (Mbps)	Max Supported # of Users
T2.NANO	1	0.5	up to 250	8
T2.MICRO	1	1	up to 250	26
T2.SMALL	1	2	up to 250	38
T2.MEDIUM	2	4	up to 500	59
T2.LARGE	2	8	2250	111
T2.XLARGE	4	16	4500	145
T2.2XLARG	8	32	9000	248

B. Transcoder as a VNF

Transcoders establishes one the major foundations of online video platforms. Any uploaded mezzanine content through CMS needs to be real time encoded to be able to support all connected screens at a time. The availability for these VNFs shows crucial importance for the success rate of the whole delivery system. However, transcoding requires considerably excessive amount of computational power due to the mathematical background of Fourier transform based processes that take place to transform spatial domain into frequency domain. Major encoding schemes mpeg4, hevc and vp9 shows different performance in terms of bitrate and storage size considering a wide range of encoding parameters.

$$M_{trans}(t) = a_M + \lambda_M . u_{Trans}(t) \frac{V_{bitrate}}{c_{encoding}} \tag{20}$$

$$C_{trans}(t) = a_c + \lambda_C . u_{trans}(t) \frac{V_{bitrate}}{c_{encoding}} \tag{21}$$

Equations 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. Obviously, for such a task, performance degradation can be crucial and ruin QoE of the whole system.

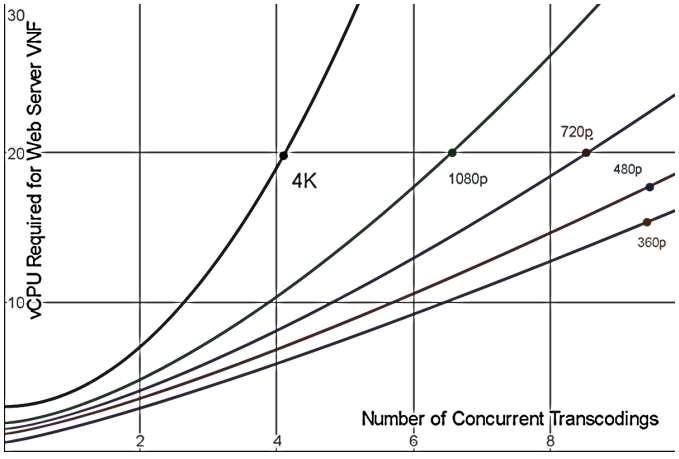

Fig. 14. vCPU required for VNF vs # of Concurrent Transcoding. The estimations correspond to physical 2.9Ghz i5 processors that are being used in AWS.

TABLE IV
CLOUD INSTANCE RESOURCE COMPARISON AGAINST MAX SUPPORTED # OF CONCURRENT TRANSCODING MAIN PROFILE HEVC 8MB 4K CONTENT

Amazon Cloud VM Standard Instance Flavors	VCPU	Memory (in GB)	MAX Supported # of Concurrent Transcoding
T4.NANO	8	16	2
T4.MICRO	8	32	4
T4.SMALL	16	64	7
T4.MEDIUM	16	128	11
T4.LARGE	32	256	14
T4.XLARGE	64	512	15
T4.2XLARG	128	512	17

VII. CONCLUSION AND FUTURE WORK

In this work, an overview of scaling strategies for online video systems have been presented with a range of comparison metrics including warming up & cooling down performance, cloud hosting costs and QoE efficiency. According to the analysis, user oriented scaling methodologies shows acceptable competence on warming up durations however their cooling down efficiency lacks the adeptness to free the underused resources when compared to resource based approaches. Throughput and computational capacity based scaling techniques shows 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, we have provided QoE scaling technique which considers all aspects of online video delivery that shows outstanding performance when compared with conventional cloud scaling strategies.

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, 2013, DOI 10.1109/TCSVT.2013.2255423

[4] V. Stocker et al, The growing complexity of content delivery networks: Challenges and implications for the Internet ecosystem, Elsevier Telecommunications Policy, 2017, https://doi.org/10.1016/j.telpol. 2017.02.004

[5] K. Mokhtarian et al, Flexible Caching Algorithms for Video Content Distribution Networks, IEEE Transactions on Networking, Canada, 2017.

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

[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, 2016.

[10] S. Joilo et al, Distributed algorithms for content placement in hierarchical cache networks, Elsevier Computer Networks, 2017. https://doi.org/10.1016/j.comnet.2017.05.029

[11] N. Kamiyama, Cache Replacement Based on Distance to Origin Servers, Ieee Transactions On Network And Service Management, Vol. 13, No. 4, 2016, DOI: 10.1109/TNSM.2016.2600240

[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), 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, Optical Society of America, 2017. https://doi.org/10.1364/JOCN.9.000053

[14] C. Rotsos et al, Network service orchestration standardization: A technology survey, UK, Elsevier Computer Standards & Interfaces, 2017.

[15] H. He et al, Dynamic Load Balancing Technology for Cloud-oriented CDN, Computer Science and Information Systems, 2015. DOI: 10.2298/CSIS141104025H

[16] *P. Frangoudis et al, CDN-As-a-Service Provision Over a Telecom Operators Cloud, Ieee Transactions On Network And Service Management, Vol. 14, No. 3, France, 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, Mexico, 2016, https://doi.org/10.1016/j.jnca.2016.05.007

[18] S. Razzaghzadeh, Probabilistic modeling to achieve load balancing in Expert Clouds, Elsevier Ad Hoc Networks, Iran 2017, http://dx.doi.org/10.1016/j.adhoc.2017.01.001

[19] J. Yue et al, Femtocaching in video content delivery: Assignment of video clips to serve dynamic mobile users, Elsevier Computer Communications, China, 2014, http://dx.doi.org/10.1016/j.comcom. 2014.05.010

[20] C. Lin, Strategy analysis for cloud storage reliability management based on game theory, Journal of Computer Security, Taiwan, 2017.

[21] Jinwei Liu, A Popularity-aware Cost-effective Replication Scheme for High Data Durability in Cloud Storage, IEEE International Conference on Big Data (Big Data), USA; 2016.

[22] Y. Tang, Achieving convergent causal consistency and high availability for cloud storage, Elsevier Future Generation Computer Systems, China, 2017.

[23] D.Kesavaraja et al, QoE enhancement in cloud virtual machine allocation using Eagle strategy of hybrid krill herd optimization, J. Parallel Distrib. Comput. (2017), http://dx.doi.org/10.1016/j.jpdc.2017.08.015.

[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, Computer Networks, Elsevier (2017), https://doi.org/10.1016/j.comnet.2017.04.039

[25] K. Bilal et al, Impact of Multiple Video Representations in Live Streaming: A Cost, Bandwidth, and QoE Analysis, IEEE International Conference on Cloud Engineering, (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, DOI 10.1007/s11227-016-1649-3

[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/