Classifier based Gateway for Edge Computing

Julius Skirelis*
*Department of Electronic Systems
Vilnius Gediminas Technical University
Naugarduko g. 41-413, Vilnius LT-03227, Lithuania
Email: julius.skirelis@vgtu.lt

Dalius Navakauskas†
† Department of Electronic Systems
Vilnius Gediminas Technical University
Naugarduko g. 41-426, Vilnius LT-03227, Lithuania
Email: dalius.navakauskas@vgtu.lt

***Abstract*—Edge Computing technology aims to replace regular cloud IoT solutions on applications where data intensity and link latency plays critical role. Improvement is achieved by placing processing at the edge of the network, deploying service close to data source of user. Limited resources of Edge devices stipulate the need to smartly distribute over devices computational tasks, as well as to implement role switching techniques in order to guarantee smooth distribution when network conditions change. Gateway technique is proposed in this paper, providing experimental comparison of Edge Computing, Cloud Computing and Content Delivery Network (CDN) data flow scenarios where terms of network delay, service time and processing time are considered. Simulation results achieved by *EdgeCloudSim* software confirms the performance gain of Classifier based Edge gateway in particular balanced hardware to load ratios.**

***Index Terms*—Edge Computing; K-Means Classifier; Cloud Computing; Cloud Delivery Network; Internet of Things.**

I. Introduction

In recent years, storing and processing all data in the cloud is associated with high expenses and high data cost as data volumes increase. Edge Computing is the opportunity for system architects to implement distibuted computing power from end to end by tapping into the capabilities of mobile field devices, gateways and cloud altogether [1]. It ensures near real time network response, allows operations offline or with intermittent connectivity to be implemented. Tasks pushed to edge of the network greatly impacts on service latency and response time. Network gateways are suitable location in communication path to perform these tasks.

Most researched Edge Computing gateways solve problem of limited hardware capabilities by adaptive scheduling. In this paper a different approach is researched – we propose adaptive most optimal link between microservice and end user selection technique based on unsupervised K-Means clusterization of links statistics and VM available resources, with further classification and decision making.

The main claim is that by the application of K-Means classifier in Edge Computing gateway it is possible to reduce end-to-end latency, with them main aim to increase user experience quality by application of classifier based stream distribution gateway on Edge Computing topology.

In this paper we present an overview of the related work, explain proposed technique and describe simulation environment, thats' investigation results are later compared in terms of network statistics values over different load.

II. Related work

Edge Computing schedulers applying Dijkstra algorithm on graph modelling network topology are widely researched [2], [3], providing complex, yet reliable task scheduling and distribution over the nodes techniques. Recent research faces Edge Computing offloading problem, where limited resources of mobile nodes or local gateways are used, however they mostly rely on maximizing response time only. Optimal VM placement on physical nodes solutions to minimize network latency between them for cloud has been proposed [4], although they rely on service and are mappable to known network infrastructure.

Fog computing shares similar demands on VM placing and scheduling problems [5], [6], but it only focuses on response time between nodes, not considering service time. Energy consumption optimization is main criteria, similarly to Edge computing, wireless sensors and gateways may be used, which is not considered in this paper. For such cases, complex solutions of genetic algorithms and meta-heuristics [7] are provided to perform gateway number optimization.

FPGA and GPU based gateways for Edge Computing are proposed [8], [9], authors emphasize use of such gateways specifically for fast image analysis and object tracking purposes. Such gateways are highly demanded for Industrial Internet of Things applications [10], because of their ability to quickly process heterogeneous sensors network. However, because of different protocols between data sources, additional preprocessing with microcontrollers or other hardware in communication paths are introduced. Such technologies are also widely researched for live video streaming and Cloud gaming applications [11].

For the estimation of delay lines and hardware resources, various adaptive classifiers like Self-Organizing Map or Multi-Layer Perceptrons are widely used [12], [13], however most popular is basic K-Means clusterization technique [14]. Because K-Means algorithm is sensitive to the initial values, improved algorithm using "shooting target" principle is proposed [15]. Authors investigating detection features from sensors in smart grid systems [16] propose training and usage of SVM classifier and state, that K-Nearest Neighbor classifier usage is improper because it needs history data (this is also true in our case).

978-1-7281-1999-1/18/$31.00 ©2018 IEEE

Thus, main objectives of this paper are: a) perform simulation of video streaming service on three different topologies: Cloud, CDN and Edge; b) incorporate K-Means classifier in process of most suitable link selection based on ping latency; c) experimentally verify proposed link selection technique.

III. Simulation

This section describes proposed technique of classifier incorporation into Edge Computing gateway, used simulation environment and varied parameters.

A live video streaming service simulation is performed in the paper. This work-flow is selected because of high demand [11] and active development by corporations like Facebook, Google or Periscope. Basic work-flow (Fig. 1) consists of single capture and multiple end user devices. One-to-many interconnection requires encoding and transcoding procedures of initial stream to satisfy criteria for different bitrates and resolutions of particular devices. Streaming is performed by sending video chunks (HLS, HDS protocols) and aiming to guarantee the best quality experience for each user while maintaining highest buffering ratio and lowest latency.

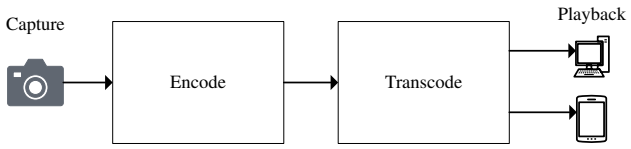

Fig. 1. Simulated service work flow

Three different topologies were simulated, each providing different path from video source to user device (see Fig. 2): Cloud based, Content Delivery Network bufferization, and Edge gateway put in communication path.

Transcoding task in the first two topologies is performed in Cloud servers, while in Edge topology, transcoding is performed locally in gateway device itself. Edge node communicates to Cloud server only with control data and state reports, main tasks are performed locally. Two techniques on Edge node are simulated and compared, resulting in 4 different topology versions being simulated.

A. Simulation environment

EdgeCloudSim simulation software together with *WEKA* K-Means clusterization library was used to perform simulation on Java virtual machine running Ubuntu linux 16.10 x64 operating system. Simulation results are saved to text files, then analysis using *MATLAB* software package is performed.

B. Distribution technique

Proposed distribution technique relies on most suitable link between Edge node and user node selection by round trip time (RTT). RTT is determined by regularly sending ICMP (ping) packet to user device. The time it takes to respond packet back represents device to node latency. To prevent network flooding by ping packets, latency query is performed each 1000-th data packet, and last 50 latency values are kept in a buffer.

Fig. 3 represents local connection diagram, where Edge node consists of infinite number of micro data centers – Virtual Machines, with only limiting factor being total hardware resources of node.

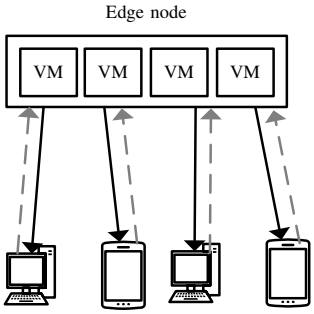

Fig. 3. Edge node internal structure and connection links

Optimal link connections for all links between Edge node and user devices are determined by performing classification. Firstly, clustering of each link latency and resources left (CPU) values by unsupervised K-Means algorithm into 3 clusters is done. Classification cloud view is presented in Fig. 4.

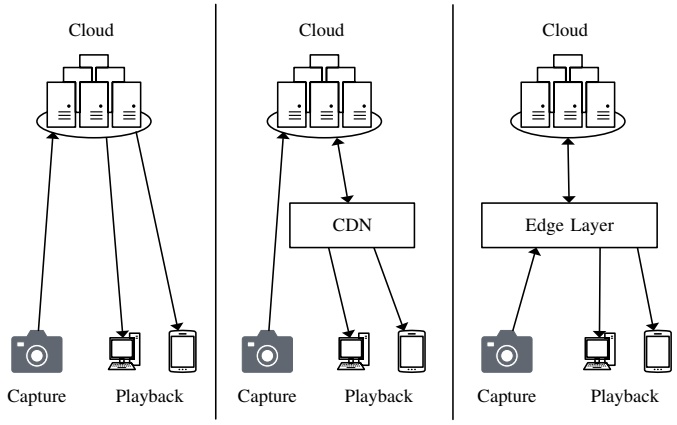

Fig. 2. Cloud, CDN and Edge topologies used for simulation

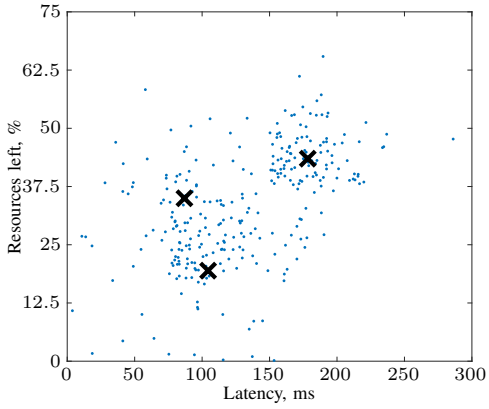

Fig. 4. View of K-Means clusterization

Later, classification is performed by assigning link to the VM that has lowest cluster index (leftmost). If VM with the lowest cluster does not establish connection in expected latency period, connection is made from next cluster. To handle situation when link may be switched to other source after each clustering operation, check for last connected source is performed to maintain connection with the same VM. Scenario when no VM can serve link is solved by opportunistic await time (in ms range) while at least one VM finish current tasks. Assignment of link to cluster is performed by calculating sum-of-squares distance to cluster centroid – this is required to match K-Means algorithm. Initial system state has no latency values collected, thus links are initially assigned in numerical increment manner.

C. Parameters

Each network topology used in simulation is defined by BRITE tool, describing nodes and edges connection rules together with network delay and maximum bandwidth parameters. Network devices are described by their resource parameters, main of them are: CPU (virtual central processing units) count, MIPS (million instructions per second) value, RAM (capacity of random access memory) and a storage value for storing buffered video stream data. For simulations, video streaming service with 5 GB data chunk size is used.

In this paper only following simulation software outputs are taken in account: network delay, processing time and service time. Network delay in our case represents amount of time for a bit of data to pass from capture device to end user device. Processing time represents a time frame between packet arrival to processing (encoding and transconding) node and outset from it. Service time is a total time data packet elapses to pass full route. All measures were evaluated as a mean value. Under small loads of ≤ 400 users, all three topologies result in constant results, to reveal advantages and disadvantages, user count in range from 400 to 1600 was selected.

For a direct Edge Computing solution comparison, a known scheduling algorithm [2] optimizing quality score was simulated. In final simulation both proposed and reference techniques use the same topology and VM constraints.

During simulation of Cloud topology, infinite M3 type VMs configuration was applied, while CDN topology was configured with up to 10 geographically distributed M4 virtual machines. Edge topology simulation was configured to allow up to 20 M1 and M2 types virtual machines to be allocated, limiting 10 instances per node. Corresponding to real hardware parameters values were used for VM configuration (Table I).

TABLE I
MACHINE PARAMETERS USED FOR SIMULATION

Name	M1	M2	M3	M4
CPU	4	8	26	2
MIPS	2400	4800	20000	2400
RAM, GB	8	16	70	16
Storage, GB	840	1680	1680	500

IV. RESULTS

Simulation results were investigated to reveal advantages and disadvantages of simulated topologies.

A. Investigation procedure

To highlight differences between used topologies, from 400 to 1600 simultaneous users are simulated as a system load. Proposed technique results are compared to Cloud, CDN and score based scheduler solutions in main user experience representing terms: mean network delay, processing time and service time accordingly. Overall service time parameter is seen as the most valuable indicator for it is worth to push video streaming related tasks closer to the Edge.

Trustworthiness of results was guaranteed by repeating each simulation 3 times and confirming the same results, that ensures no error is introduced by random generated initial variables in simulation software internally.

B. Performance analysis

First measure in analysis – mean network delay, its values over different topologies are provided in Fig. 5. It can be seen, that results are pretty constant, indicating that increase of load on system does not affect mean network delay value. However different topologies has different value because of different physical length between nodes, as it affects data bit propagation time dramatically.

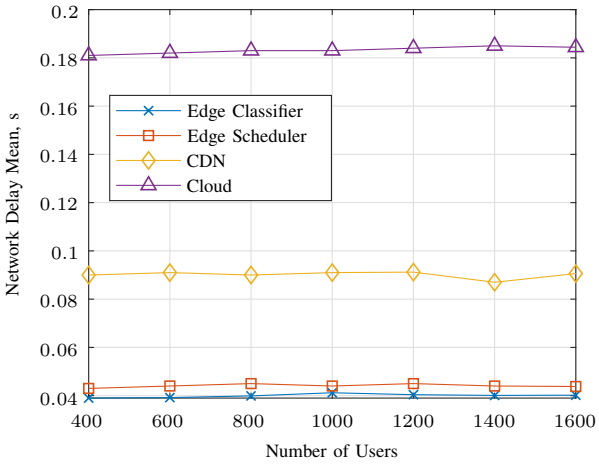

Fig. 5. Mean network delay over number of users for different topologies

Results on mean service time (see Fig. 6) shows constant values over whole range for Cloud topology, while hardware resources limited topologies reach their throughput limits at 1200 simultaneous users. It is seen how K-Means classifier initially performs worse than scheduler based approach, but later, when history ping entries are filled, it improves.

Results on processing time (Fig. 7) are expected to be similar to service time trends, since they are derived measures, although absolute mean processing time values differs dramatically over topologies. It is seen that Cloud solution itself has best processing time, but considering very limited

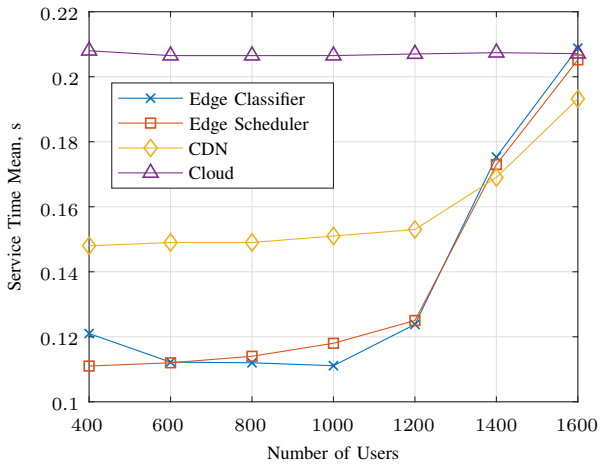

Fig. 6. Mean service time over number of users for different topologies

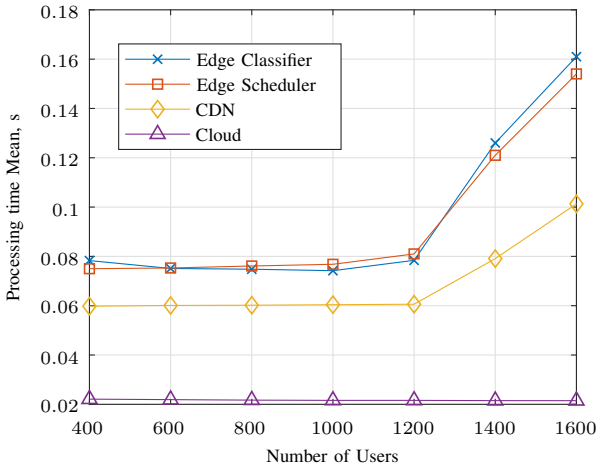

Fig. 7. Mean processing time over number of users for different topologies

hardware resources provided for CDN and Edge nodes, they are performing quite well, until bottleneck is reached.

To summarize, Cloud topology shows least processing time and stable network delay together with service time because of static network characteristics as well as high resources. Placing resource limited nodes close to the Edge results in greatly improved service time and network delay, which are main parameters representing overall user experience in live video streaming service.

V. Conclusions

Simulation of classifier based link selection technique to Edge Computing Gateway using *EdgeCloudSim* software package was performed. Performance analysis results show that:

1) K-Means classifier based link selection technique outperforms score based scheduler algorithm in range from 600 to 1200 simultaneous users in mean service time;
2) Edge Computing topology based techniques reduces mean network delay and service time by more than 3 times;
3) loading Edge network with more than 1200 simultaneous users results in hardware bottleneck, therefore processing time and service time starts to increase exponentially;
4) initial run of classifier based network results in twice as higher service time compared to scheduling algorithm, that later settles down because latency values of all nodes becomes available.

References

[1] J. Skirelis and D. Navakauskas, "Edge computing in iot: Preliminary results on modeling and performance analysis," in *5th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*. IEEE, 2017, pp. 1–4.

[2] V. Scoca, A. Aral, I. Brandic, R. De Nicola, and R. B. Uriarte, "Scheduling latency-sensitive applications in edge computing." in *Closer*, 2018, pp. 158–168.

[3] M. Jridi, T. Chapel, V. Dorez, G. Le Bougeant, and A. Le Botlan, "Soc-based edge computing gateway in the context of the internet of multimedia things: Experimental platform," *Journal of Low Power Electronics and Applications*, vol. 8, no. 1, p. 1, 2018.

[4] A. Aral and T. Ovatman, "Network-aware embedding of virtual machine clusters onto federated cloud infrastructure," *Journal of systems and software*, vol. 120, pp. 89–104, 2016.

[5] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards QoS-aware Fog Service Placement," in *2017 IEEE 1st international conference on Fog and Edge Computing (ICFEC)*, 2017, pp. 89–96.

[6] V. B. Souza, A. Gomez, X. Masip-Bruin, E. Marin-Tordera, and J. Garcia, "Towards a Fog-to-Cloud Control Topology for QoS-Aware End-To-End Communication," in *2017 IEEE/ACM 25th international symposium on Quality of Service (IWQOS)*, 2017.

[7] I. Gravalos, P. Makris, K. Christodoulopoulos, and E. A. Varvarigos, "Efficient Network Planning for Internet of Things With QoS Constraints," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3823–3836, 2018.

[8] A. Rodriguez, J. Valverde, J. Portilla, A. Otero, T. Riesgo, and E. de la Torre, "Fpga-based high-performance embedded systems for adaptive edge computing in cyber-physical systems: The artico3 framework," *Sensors*, vol. 18, no. 6, p. 1877, 2018.

[9] Z. Zhao, Z. Jiang, N. Ling, X. Shuai, and G. Xing, "Ecrt: An edge computing system for real-time image-based object tracking," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2018, pp. 394–395.

[10] C.-H. Chen, M.-Y. Lin, and C.-C. Liu, "Edge computing gateway of the industrial internet of things using multiple collaborative microcontrollers," *IEEE Network*, vol. 32, no. 1, pp. 24–32, 2018.

[11] K. Bilal and A. Erbad, "Edge computing for interactive media and video streaming," in *Second International Conference onFog and Mobile Edge Computing (FMEC)*. IEEE, 2017, pp. 68–73.

[12] L. Stasionis and A. Serackis, "A new method for adaptive selection of self-organizing map self-training endpoint," *Baltic Journal of Modern Computing*, vol. 3, no. 4, p. 294, 2015.

[13] D. Plonis, A. Katkevičius, V. Urbanavičius, D. Miniotas, A. Serackis, and A. Gurskas, "Delay systems synthesis using multi-layer perceptron network." *Acta Physica Polonica, A.*, vol. 133, no. 5, pp. 1281–1286, 2018.

[14] G. Li, S. Xu, J. Wu, and H. Ding, "Resource scheduling based on improved spectral clustering algorithm in edge computing," *Scientific Programming*, vol. 2018, pp. 1–13, 2018.

[15] Z. Wang, K. Wang, S. Pan, and Y. Han, "Segmentation of Crop disease images with an improved K-Means clustering algorithm," *Applied engineering in agriculture*, vol. 34, no. 2, pp. 277–289, 2018.

[16] P. Xun, P. Zhu, Z. Zhang, P. Cui, and Y. Xiong, "Detectors on Edge Nodes against False Data Injection on Transmission Lines of Smart Grid," *Electronics*, vol. 7, no. 6, pp. 1–12, 2018.