GPU-accelerated Video Transcoding Unit for Multi-access Edge Computing Scenarios

Antonino Albanese, Paolo Secondo Crosta, Claudio Meani, Pietro Paglierani,
ITALTEL,
Castelletto, Milan, Italy

e-mail: {antonino.albanese, paolosecondo.crosta, claudio.meani, pietro.paglierani}@italtel.com

***Abstract*—The exponential growth of video traffic and the outburst of novel video-based services is revealing the inadequacy of the traditional mobile network infrastructure. To respond to this and to many other demands coming from today's society, the 5G and the Multi-access Edge Computing (MEC) initiatives are proposing novel network architectures. In this context, this paper proposes the Video Transcoding Unit (VTU) application, which, leveraging on MEC principles, brings several functionalities to the edge of networks, greatly improving User Experience with mobile terminals. The VTU can be implemented as a SW (Software)-only Virtual Network Function, or be accelerated by a Graphics Processing Unit (GPU). Specific tests are described and discussed, showing the clear superiority of the HW (Hardware)-accelerated implementation in terms of computing performance and efficiency. A possible use case is presented, in which the VTU is used in a Stadium or in large public venues during crowded events like a sporting match or concerts. The work presented in this paper was undertaken under the EU Horizon2020 Sesame Project.**

Keywords-NFV; MEC; 5G; HW-acceleration; GPU; Video transcoding.

I. INTRODUCTION

The recent worldwide explosion of mobile data traffic has been impressive, and it is clear that this trend will continue in the coming years.

The fast spreading of smart terminals together with new services based on high-definition video have been the main triggers of this explosion, revealing the inadequacy of the architectural and technological approach adopted so far in the design of the traditional mobile network infrastructure. The telecommunication market, previously dominated by voice traffic and text messages, is rapidly shifting to a completely different and far more complicated scenario, made of millions of connected applications where even different actors have made their appearance, like machines and "things" (smart home gadgets, vehicles, drones, robots, also including sensors and actuators).

This way, Internet and communication networks have become crucial for any evolutionary process of modern societies and economies. This fact led to the definition of a new kind of infrastructure based on the "fifth generation" - 5G - architecture as a response to the requirements coming from the more diverse fields of the future world [1].

5G aims at assuming a fundamental role in the new society; it is not only a simple evolution of previous mobile networks – as was the passage from 3G to 4G - but it stands as a real revolution, able to create the appropriate ecosystem for technical and business innovation [2].

From the technological point of view 5G will take advantage of the last years' experience coming from the convergence of the telecom world with Information Technology. This strong movement addressed the necessity coming from Network Operators of reducing general costs, achieving better scalability and reducing the deployment time of new services and resulted in a new architectural vision based on Software-Defined Networking (SDN) and Network Function Virtualization (NFV) [3].

5G will bring the SDN and NFV concepts in the radio communications environments and will use them in a new architectural framework where Multi-access Edge Computing (MEC) will play a major role.

MEC Technology and Architecture concepts are a way to improve both Efficiency and User Experience for a certain number of services. MEC is an ETSI initiative that uses virtualization, small cells, SDN and NFV principles to push network functions, services and content to the edge of the mobile network [4][5].

The MEC servers are typically directly attached to the base station, but this is not a strict rule because, in this regard, the MEC guidelines are widely open. They provide computing, storage and networking resources that are virtualized and shared by multiple virtual machines.

Traditionally, all data traffic originating in data centres is forwarded to the mobile core network. The traffic is then routed to a base station that delivers the content to the mobile devices. In the mobile edge scenario, MEC servers take over some or even all of the tasks originally performed in a data centre. Being located at the mobile edge, this eliminates the need of routing data through the core network, lowering communication latency. As such, the MEC paradigm helps to reach the severe requirements posed by 5G in terms of throughput, latency, scalability and automation. It's important to note that many of the concepts that are at the basis of MEC and the advantages they bring to a broad range of services are valid regardless of 5G technology (in fact MEC concepts can be similarly applied to fixed networks) and can be demonstrated prior to the coming 5G.

There are many services that could benefit from being hosted at the edge of the network. Several use cases have been defined in the specification of MEC architecture to demonstrate the advantages of the introduced concepts. One of these use cases regards video traffic in stadiums and/or large public venues where the video created during a sport event or a concert is routed to a MEC server that is responsible for its local distribution, without involving backhaul connection to the core network. The video contents are then stored in this edge platform and can be locally

 ISBN: 978-1-61208-546-3

elaborated with applications running on the same MEC server to create new services and improve User Experience.

This paper presents the Italtel VTU application, which, leveraging on MEC principles, brings several functionalities to the edge of networks, greatly improving User Experience with mobile terminals. VTU speeds up upload and download of Video contents, reduces latency and contribute to increasing the battery life of connected devices offloading them from heavy transcoding operations.

This paper shows how an application such as the VTU could fit in a real use case foreseen by 5G and MEC and what are the limitations of a SW-only implementation with respect to a GPU-accelerated one.

The paper is organized as follows. Section II provides a brief functional description of the VTU. Section III briefly discusses why HW acceleration should be considered in developing a VNF such as the VTU. Section IV presents the performance characterization of VTU with and without GPU. Section V shows a possible use case for VTU during localized crowded events. Finally, Section VI summarizes the main results of this work.

II. VTU DESCRIPTION

The VTU can convert video streams from one video format to another. It can either run on bare metal environments, or it can be implemented as a VNF providing optimized video transcoding function, for the benefit of many other VNFs, to create enhanced services.

Depending on the type of application that should be provided, the source video stream could originate from a file within a storage facility, as well as coming in form of packetized network stream from another VNF. Moreover, the requested transcoding service could be mono-directional, as in video stream distribution-like applications, or bi-directional, like in videoconferencing (see Figure 1).

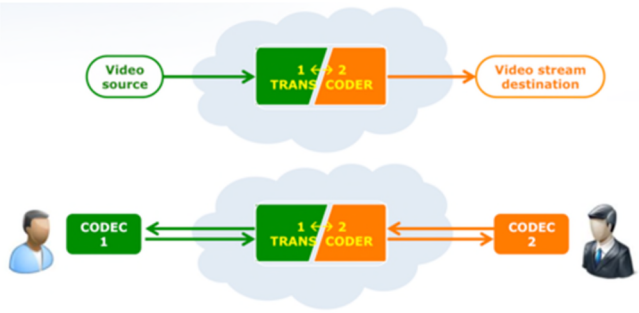

Figure 1. Simplified VTU model .

In the VTU, the audio and video transcoding capabilities are provided by the Libav library [6], a very popular open source library, which can perform encoding and decoding according to a wide set of coding standards. The AVConv tool from Libav is used for performing the conversion between audio and video formats and containers; while it already supports a wide variety of hardware accelerations, native GPU support in encoding tasks is quite limited, experimental and restricted only to H.264 and H.265 standards, exploiting the NVidia NVENC hardware encoder of medium and high level NVidia GPUs [7]. The AVConv tool running in the VTU has been modified so that VP8 encoding tasks can also greatly benefit from the virtualization of GPUs.

The VTU VNF is implemented as SW module running on a virtual machine and can be installed in one or multiple physical servers clustered together through a local communication fabric. The VTU can support a large set of video codecs, and in particular the most recent and popular ones, such as H.264, H.265 and VP8/VP9 [8][9].

III. VTU AND HW ACCELERATION

Although software-only functions can give acceptable performance in many applications, when compute-intensive workloads running at the data plane are of interest, such as those based on video data processing, quite poor results can be obtained. In these cases, to reach the expected performance, it is often necessary to consider a slightly different approach that involves the use of Hardware accelerators. In general, managing HW accelerators and making them transparently available to every VNF goes against the assumption of every virtualized environment, of having a uniform HW platform made of CPU-only computing elements. The presence of HW accelerators bound to a virtual function implies the use of a SW layer that must be HW-aware, thus significantly complicating system management operations and scalability [3][9]. Though, the advantages of HW acceleration can be so preponderant, in particular when performance, latency or Service Level Agreement (SLA) requirements are challenging, that not considering them can push a commercial product out of the market. In fact, acceleration is not just related to performance, but also to the reduction of the number of physical servers, footprint, network appliances and power consumption. In short, it can make the difference in the commercial proposition of a product.

A distinctive feature of VTU is the possibility not only to run on general purpose CPUs but also to exploit the Hardware acceleration provided by a GPU, to improve the compute performance of video codecs. To this end, two different architectural approaches can be used. The first one, also known as “cooperative CPU- GPU” makes use of a GPU to offload the most compute-intensive functions of the video codec (usually, the Motion Estimation block), while the main algorithm is kept running on the CPU. The second approach, conversely, uses full HW implementation of video codecs. Today, various HW versions of the most popular encoding schemes, such as H.264, HEVC, VP8 and VP9, are available [7][10]. The fully HW approach can provide higher compute performance than cooperative CPU-GPU algorithms. Though, the HW approach very often lacks the flexibility in service management needed by service operators, thus the cooperative approach is still preferred in many real-life implementations. The VTU can adopt both GPU-accelerated approaches. In fact, it can use the Nvidia NVEnc encoder for the H.264 and H.265 encoding schemes [7]. Also, the CPU-GPU cooperative approach described in [11][12] can be used for the Google open Source VP8 encoder.

 ISBN: 978-1-61208-546-3

IV. VTU PERFORMANCE

We carried out many tests in Italtel laboratories to achieve a full performance characterization of the VTU, both for the SW-only version, and the GPU-accelerated one. For the sake of brevity, in the following, a few meaningful results are presented and discussed. In particular, Figure 2 and Figure 3 show the results obtained with the VTU featuring the H.264 and H.265 transcoding, (expressed in frames per second) without HW acceleration (SW-only) and with HW acceleration (using a GPU). The processing implies decoding from the input format to the one required as output.

In all tests, the same H.264 Full HD video file (1080x1920 resolution) was used as input. The VTU provided four different video resolutions as output, in four different transcoding tests: VGA (480x640 pixel), HD480 (480x852 pixel), HD720 (720x1280 pixel), HD1080 (1080x1920 pixel).

The horizontal axis in Figure 2 and Figure 3 represents output resolution, while the vertical axis indicates the achieved output frame-rate in frames per seconds (fps).

The SW-only VTU was running on a server with a single socket Intel Xeon E5-2630v3 2.4GHz, 8 Core CPU, with 64 GB DDR4 RAM. The Encoder used in this case for H.264 and H.265 is X264 and X265, respectively.

The GPU-accelerated VTU was running on the same server with the addition of a NVIDIA® QUADRO® M4000 GPU in a x16 PCIe Slot. The Encoder used by the GPU for H.264 and H.265 is NVIDIA NVENC.

Figure 2 and Figure 3 collect the results of the tests achieved with H.264 and H.265 encoders respectively. In both cases only one session was launched for each test.

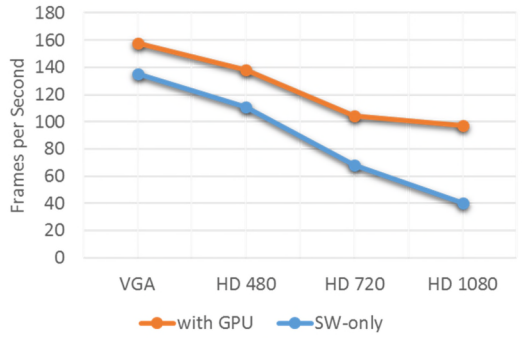

Figure 2. H.264 single session encoding performance (higher is better).

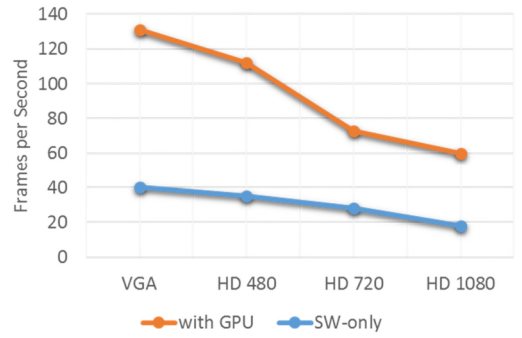

Figure 3. H.265 single session encoding performance (higher is better).

As one can easily see, the performance improvement using the GPU compared to a SW-only solution is remarkable in all cases. This confirms the need of GPU acceleration especially in modern and future scenarios where 4K or even higher video resolutions are going to be used.

Another important aspect to emphasize is related to the occupation of compute resources during transcoding. Although in SW-only mode CPU resources were completely occupied (all the CPU cores were running at 100%), using the GPU both CPU and GPU resources were only partially used. For example, during the H.264-HD1080 test reported before, in which only one encoding session was launched, the VTU was using only 20% of available GPU resources. This fact led us to a second set of tests in which multi-session performance was analyzed. In this new set of tests, the focus was on a single case, i.e., H.264 HD1080, launching 2, 4, 8, and 16 concurrent transcoding sessions.

The results are reported in Figure 4 and Figure 5.

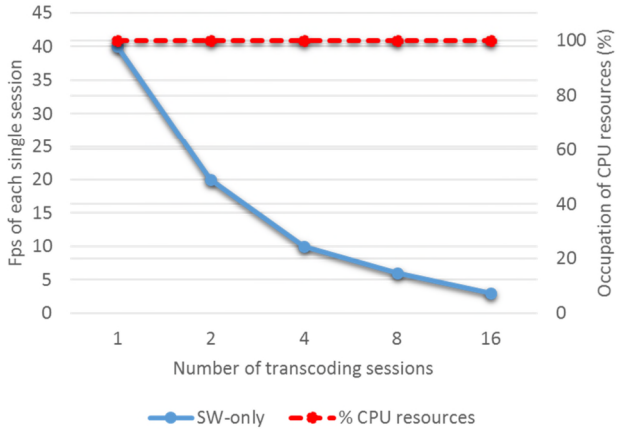

Figure 4. H.264 HD1080 encoding SW-only in multi-session transcoding tests (performance are related to each single session) with percentage of CPU resources utilization.

Considering the SW-only implementation (Figure 4) the performance of each single session decreases with the total number of executing sessions. Comparing the performance of 1 session to that with 16 concurrent sessions the result is the same. In fact, aggregating the fps of all the 16 sessions we obtain 16x2.5= 40fps (in case of 16 concurrent sessions the single session fps is 2.5). This can be easily justified considering that the CPU occupation during the processing is always around 100% also running a single session.

The same is not true using the GPU (Figure 5). In this case the CPU is only partially used because the workload is mainly offloaded to the GPU whose resources are, in turn not fully used (as the dotted lines show). Using the GPU with 16 concurrent sessions we reach 24 fps for each session, for a total of 16x24=384fps. The 384/40 ratio brings to a 9.6x gain in performance using the GPU respect to a SW-only solution. During the GPU test with 16 transcoding sessions the CPU was running at 70% giving it the possibility to run other tasks. This was not possible with SW-only solution, because in such a case the CPU was always 100% occupied.

 ISBN: 978-1-61208-546-3

Comparing the efficiency of the two solutions in term of performance/watt we see another important advantage of using the GPU. In fact, in case of 16 sessions (H.264-HD1080), the power consumption of the server running the SW-only VTU is around 200W while with GPU is around 300W. The gain in efficiency for GPU-accelerated VTU is then 6.4 ((384/300) / (40/200)).

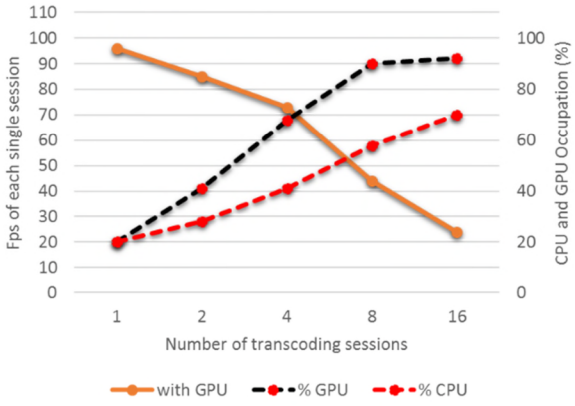

Figure 5. H.264 HD1080 encoding with GPU in multi-session transcoding tests (performance are related to each single session) with percentage of CPU and GPU resources utilization.

Similar considerations could be made regarding costs (the GPU used costs only around 70% the price of the server) and physical space (the space occupied by the server with or without GPU is the same, the latter being hosted inside the server).

V. A USE CASE FOR VTU

A possible use case for the VTU can be described by the following two scenarios.

"Imagine being at a stadium, where a football match takes place. Your team scores a goal but you are not in the best position to appreciate it or the action was confusing and you did not realize who scored the goal and how. You would like to have the possibility to watch on your smartphone the most relevant actions from different points of views"

Or:

"You are attending a crowded concert in the front row close to the stage and you want to show to other friends attending the concert far from the stage some video in real time, picturing the performance in progress. Also, the concert organizers could decide to show on the gigantic main screen a collage of real time videos coming from spectators to give them a more immersive and engaging experience."

In this type of contexts, there is an overwhelming demand for services that give the possibility to the users to have videos on their smartphones or tablets on demand, as services provided, for instance, by the Stadium.

From the technological perspective, what is needed to implement such type of services is a networking infrastructure featuring a very rapid upload and download of large files, such as HD videos. In addition to that, the possibility to process in a highly effective way video streams is a mandatory function to provide enhanced services. VTU, implemented in a MEC environment, represents a possible answer to that demand coming from the market. The HW-accelerated transcoding of video streams can help in reducing the computational workload of mobile terminals converting video streams from the uploaded format to one more suitable for the receiving terminal, increasing its battery life.

The whole process of upload, transcoding and download takes place locally in the MEC server (Figure 6) offloading the backhaul connection towards the core network. This reduces latency and avoids backhaul traffic congestion.

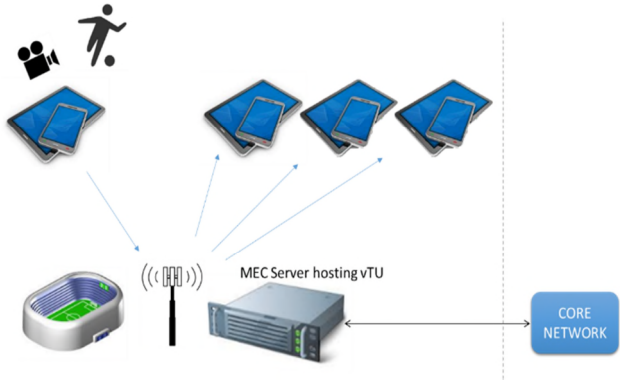

Figure 6. VTU use case: providing low lantency video sevices during localized crowded events leveraging MEC architecture

To this end, the MEC server must be equipped with its own high performance storage where all the videos uploaded from the users are kept for a certain amount of time, for instance a week. During this period a suitable application can make them available on demand outside the perimeter of the stadium, e.g., at home. The spectators during a sporting event can then upload many videos and delete them immediately to preserve memory space on their mobile devices, having the possibility to choose at a later time which one to download.

To provide these services, the Stadium or the event organization will make available an App to download on spectators' smartphones.

VI. CONCLUSION

This paper has presented the Video Transcoding Unit (VTU) application, which, leveraging on MEC principles, brings several video data processing functionalities to the edge of networks, greatly improving User Experience with mobile terminals. The VTU can be implemented as a SW-only VNF, or be accelerated by a GPU. Specific tests have been reported showing the clear superiority of the HW-accelerated implementation. A possible use case has been presented in which the VTU is used in a Stadium or in large public venues during crowded events like a sporting match or a concert. This work was undertaken under the EU Horizon2020 Sesame Project.

 ISBN: 978-1-61208-546-3

ACKNOWLEDGMENT

This research received funding from the European Union H2020 Research and Innovation Action under Grant Agreement No.671596 (SESAME project).

The authors are grateful to Mr. Marco Beccari and Mr. Luca Di Muzio who carried out the laboratory tests described in this paper.

REFERENCES

[1] 5G Infrastructure Public Private Partnership (PPP): The next generation of communication networks will be Made in EU. Digital agenda for Europe. Technical Report, European Commission. February 2014.

[2] NGMN: 5G White paper (2015).

[3] ETSI: ETSI GS NFV-MAN 001 v1.1.1: Network Functions Virtualisation (NFV); Management and Orchestration (2014)

[4] ETSI: Mobile-Edge Computing - Introductory Technical White Paper (2014)

[5] B. Blanco et al., “Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN”, Computer Standards & Interfaces, Available online 4 January 2017, ISSN 0920-5489.

[6] Libav. [Online]. Available from: http://libav.org/documentation/ 2017.03.27

[7] NVIDIA NVENC Programming Guide [Online]. Available from: https://developer.nvidia.com/nvenc-programming-guide 2017.03.27.

[8] N. M. Cheung, X. Fan, O. C. Au, and M. C. Kung, "Video Coding on Multicore Graphics Processors," in IEEE Signal Processing Magazine, vol. 27, no. 2, pp. 79-89, March 2010.

[9] P. Paglierani, "High Performance Computing and Network Function Virtualization: A major challenge towards network programmability," 2015 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Constanta, 2015, pp. 137-141.

[10] WebM Video Hardware RTLs [Online]. Available from: https://www.webmproject.org/hardware/ 2017.03.27.

[11] P. Comi et al., "Hardware-accelerated high-resolution video coding in Virtual Network Functions," 2016 European Conference on Networks and Communications (EuCNC), Athens, 2016, pp. 32-36.

[12] P. Paglierani, G. Grossi, F. Pedersini, and A. Petrini, "GPU-based VP8 encoding: Performance in native and virtualized environments," 2016 International Conference on Telecommunications and Multimedia (TEMU), Heraklion, 2016, pp. 1-5.

 ISBN: 978-1-61208-546-3