Wide Area Video Surveillane Based on Edge and Fog Computing Concept

G.Kioumourtzis and M.Skitsas
Advanced Integrated Technology Solutions & Services
ADITESS Ltd.
Nicosia, Cyprus
{research, mskitsas}@aditess.com

N. Zotos and A.Sideris
Future Intelligence Ltd.
London, UK
{nzotos, asideris}@f-in.co.uk

***Abstract*— Current Surveillance systems in enterprise facilities and public places produce massive amounts of video content while operating at a 24/7 mode. There is an increasing need to process, on the fly, such huge video data streams to enable a quick summary of "interesting" events that are happening during a specified time frame in a particular location. Concepts like fog computing based on localisation of data processing will relax the need of existing cloud-based solutions from extensive bandwidth and processing needs at remote cloud resources. In this paper, we describe a novel architecture for a smart surveillance system based on edge and fog computing concepts. We provide the main architectural components, the hardware options and key software components of the system. Edge computing is realized by a camera embedded system while fog computing is used for the processing and data fusion of video streams in small areas. Lab tests concentrated on the different versions of edge computing devices have shown the efficiency of the system.**

***Keywords*— surveillance system; edge computing; fog computing; Video content management**

I. Introduction

Current Surveillance systems in enterprise facilities and public places produce massive amounts of video content while operating at a 24/7 mode. Typically, video clips are stored as compressed video files corresponding to video for several hours. Nowadays, there is an increasing need to process, on the fly, such huge video data streams to enable a quick summary of "interesting" events that are happening during a specified time frame in a particular location. For instance, one might be interested in anomalous trajectories of objects within a scene. This is mainly related to the extraction of "useful data" for Law Enforcement Agencies (LEAs) either for situational awareness or for intelligence analysis.

Due to the prohibitively high requirements of surveillance video and the need for continuous monitoring, currently closed-circuit television (CCTV) systems store video on local devices. Upon the occurrence of an event, one needs to visit the location of the incident and retrieve the content manually and then proceed by either watching the full length of the video for the identification of the footage of interest or processing through specialised video analytics algorithms. Cloud infrastructure solutions may involve specialised algorithms and methods for the distribution of the required processing load to increase efficiency in terms of time. Often, the situation becomes more complicated when other types of data are involved in the processing (e.g. feeds from social media, other video sources like YouTube, Periscope, sensor systems, etc.). Video analytics nowadays receive as input HD or Full HD video, increasing substantially the strain on the hosting infrastructure for real time processing and provision. Even though H.264 achieves higher compression rate compared to MPEG-4, video remains a very resource hungry type of multimedia. This is further emphasised by the figures shown in TABLE I, where within the duration of one hour, 1000 video streams in a video surveillance infrastructure generate tens of terabytes of data that need to be transferred over the network for intelligent video surveillance analysis.

TABLE I. STORAGE REQUIRED FOR 1HOUR VIDEO PRODUCED BY 100 STREAMS

Frames per second	Resolution	Encoding	Storage
30	1240 x 1024	MPEG-4	125 TB
30	704 x 480	MPEG-4	35.75 TB
30	1240 x 1024	H.264	71.5 TB
30	704 x 480	H.264	22.75 TB

On the other had new technological solutions and infrastructures offer unique opportunities for smarter and more efficient surveillance. Cloud computing is "the infrastructure" for real-time data-intensive applications due mainly to its theoretically boundless scalability power. However, in dynamic IoT-based and Big Data driven environments, there are still valid challenges with respect to efficient and reliable real-time processing, since the existing widely adopted technologies (e.g. Hadoop) are not applicable for real-time processing due to their static nature [1]. In addition, although more recent and dynamic technologies like STORM, offer a very efficient computational solution for real-time processing, they don't exploit new decentralised paradigms (e.g. distributed clouds, edge [2] and fog computing [3].

Nevertheless, there are arguments that highlight as a major potential downside of cloud computing the slower-than-desirable performance due to networks' bandwidth limitations [4]. The problem of how to process efficiently on the cloud is becoming more acute as more and more objects become "smart," or able to sense their environments, connect to the Internet, and even receive commands remotely. Modern 3G and 4G cellular networks simply aren't fast enough to transmit data from devices to the cloud at the pace it is generated. The above facts led the research community to new concepts on fog computing since they store and process the torrent of data being generated by the Internet of Things (IoT) on the "things" themselves, or on devices that sit between the "things" and the Internet. According to Cisco [3], the fog extends the cloud to be closer to the things that produce and act on IoT data. These devices, called fog nodes, can be deployed anywhere with a network connection: on a factory floor, on top of a power pole, alongside a railway track, in a vehicle, or on an oil rig. Any device with computing, storage, and network connectivity can be a fog node. Examples include industrial controllers, switches, routers, embedded servers, and video surveillance cameras. A more specific example involves Cisco selling routers, which aside from storage, they offer additional and competitive to the market business added value, by turning its routers into hubs for gathering data and making decisions based on them. IDC estimates that the amount of data analysed on devices that are physically close to the Internet of Things is currently approaching 40 percent [5]. This is quite reasonable, since latency reduction for quality of service (QoS) and edge analytics/stream mining, result in superior user-experience [6] and redundancy in case of failure ([7], [8]).

Driven by the latest developments and facts we present in this work a novel surveillance system based on fog and edge computing. Our solution involves small computing devices at the sensor level (camera) that provide a first level of processing based on "light" video and audio analytics. The various video streams from the same location (e.g. building) are fused to fog nodes where a second level processing is taking place. A private cloud infrastructure with higher computing and processing capacity can be utilised to cope with unpredicted workload of a diverse number of video streams.

In the next Section, we present the architecture of the surveillance system while in Section III we discuss the hardware and software components of the camera embedded system. The fog node is presented in Section IV. Lab tests are presented in Section V. Finally, we conclude our work in Section VI.

II. Wide Area Video Surveillance System Architecture

CCTV technology is a measure for increasing security. In modern European cities, there are several CCTV cameras installed in public facilities and buildings, shopping malls and traffic lights. Some of these cameras are old analogue while others are new IP-based digital cameras that offer high quality video and easier access. Security personnel in several facilities watch video streams from several cameras in control rooms to identify security threats and notify Police.

A. *Problem Definition and Challenges*

So far, the deployment of CCTV cameras has proven to be non-efficient due mainly to the following factors:

- Current systems are coming from different manufacturers. The lack of adoption of widely accepted technical standards makes them incompatible and non-interoperable.
- Currently, there is still a need to physically transport video surveillance footage while the operator manually analyzes the video.
- Video streaming is a bandwidth hungry application area.
- From the operational perspective, it is hard for security personnel to monitor many security cameras simultaneously.
- The response to security threats is low with current architectures, as many human factors are involved in the chain from event detection to Police notification.
- Citizens' sense of security remains low due to poor results of the existing CCTV cameras and the poor performance of the whole system.

B. *Possible Solutions*

To address the identified gaps and make existing CCTV surveillance cameras more efficient, a novel architecture is proposed with a focus on the use of fog and edge computing concepts. With this new architecture, surveillance systems can cover more efficiently all sensitive areas in small towns and larger cities including public administration buildings, bus stations, shopping malls, main squares and any other sensitive area.

The application logic of the new architecture is to gather video streams from the cameras, store and process video clips in the case of identified threats, for instance:

- Perimeter protection
- No trespassing of security areas
- Gunshots
- Screaming

The primary detection of these types of security threats is based on a camera-built embedded system with computing and storage capacity. For more efficient handling and processing, the embedded systems are connected to Fog Nodes installed in central locations (e.g. building, shop malls, train stations etc.) that will provide additional computing and storage resources. The Fog Nodes can be connected to a public or private cloud, when needed, to make use of additional computing resources to cope with unpredicted workload at the Fog Node level and provide costumer-tailored business intelligence.

Control rooms in each one central location are equipped with video walls where security personnel can watch on real time the video streams coming from the cameras. In the case of an identified threat, an alarm is triggered and security personnel can notify the Police for response.

Therefore, we can distinguish the following processing layers, as follows:

- Layer-1: Camera-built embedded system

At this level, all camera-built embedded systems under the same or different domains will be synchronised based on localisation of data processing.

- Layer-2: Fog Node

At this level, all Fog Nodes will operate under a common monitoring service where computing and network resources will be shared amongst them for higher performance.

- Layer-3: Public or Private Cloud

At this level service migration between Fog Nodes and the cloud will be realised when Fog Nodes are not able to handle sudden picks in the workload.

An overview of the proposed technology for the new surveillance system is depicted in Fig. 1. We can distinguish three major elements a) the camera-built embedded system, b) the Fog Node Unit, and c) the public or private cloud.

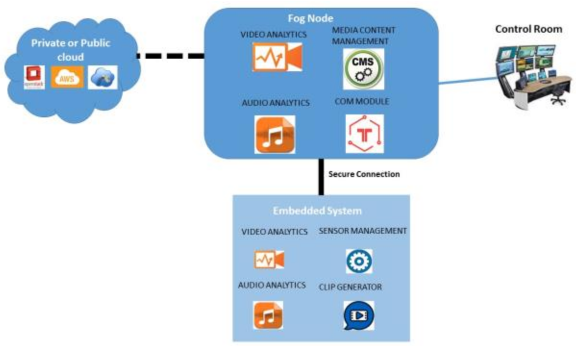

Fig. 1. Survaillance system architecture.

III. CAMERA EMBEDDED SYSTEM (ES)

The 1st Layer of the smart wide area surveillance system is the embedded system with the Camera. The embedded system has designed and developed in order to handle:

- Sensor management tasks. Connectivity and information retrieval from different cameras and microphones.
- Video processing in order to achieve unified video format.
- Lightweight video analytics (e.g. intrusion detection).
- Intelligent management of Lightweight analytics and embedded system operations (storage efficiency, network adaptation etc.).

The embedded system is capable of interconnecting various types of cameras (USB, Network cameras etc.). It is very important to mention that the embedded system has two modes: i) smart operation, and ii) traditional CCTV. As the 1st layer of the proposed system the embedded system has the capability of performing lightweight identification of various events. The fog nodes can perform more complicated analysis and processing. The main building blocks comprising the embedded system are as follows:

A. *Hardware Component*

The ES builds on a low cost (i.e. approximately 45 Euros), small size (84x60 mm), dual core (e.g. Cortex-A7 processor, A20-OLinuXIno-LIME2 board [9]). The board provides 1GB RAM and a fast Ethernet connection of 1 Gbps. Furthermore, the existence of two USB-2.0 ports enables not only the use of USB based sensors (e.g. cameras) but also the utilisation of USB WIFI dongles for connecting to wireless sensors, a feature that eases the deployment of the Embedded Systems in the field. A setup of this system, purposed for indoor environments, is depicted in Fig. 2.

Fig. 2. FINT Ltd. embeded system prototype with network camera.

The ES can be utilized for running video and audio analytics based on end-user needs. The ES can provide a first notification of any abnormal detection at a local level.

When any type of the aforementioned security threats is detected, then video is streamed back from the camera to the respective fog node for further identification and verification.

In this context, the first high-end ES, namely the ODROIDXU3 Lite [10] board, was equipped with an Exynos-5422 application processor that builds on two quad-core central processing units, namely the CortexTM-A15 and the CortexTMA7, and a GPU (Mali T628) with computing capabilities. In addition, the board provides 2GByte (DDR3) RAM, a feature that enables the analytics to keep more video data in the faster RAM memory than retrieving them from the boards SD card, where the data are typically stored. It is noted that all this processing power comes to the expense of the power efficiency (less autonomy), but nevertheless a trade-off should be expected when a more sophisticated analysis comes to the table.

B. Software Architecture

This section describes the software modules that deployed in the ES prototype. The ES pre-processes the incoming video stream; clips the stream in short video files, named Clip Data; feeds the stream to video analytics; collects the analytics' results; evaluates them; and if the evaluation outcome indicates a possible anomalous event, it creates and sends the video clip to the respective Fog Node. The Clip is a key information unit that contains both the Clip Data files, containing the detected event, and a metadata file, named Clip Object.

The embedded system is consisted of the following modules (Fig.3).

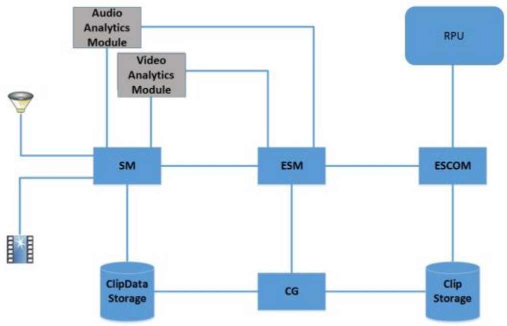

Fig. 3. Embedded system software architecture diagram.

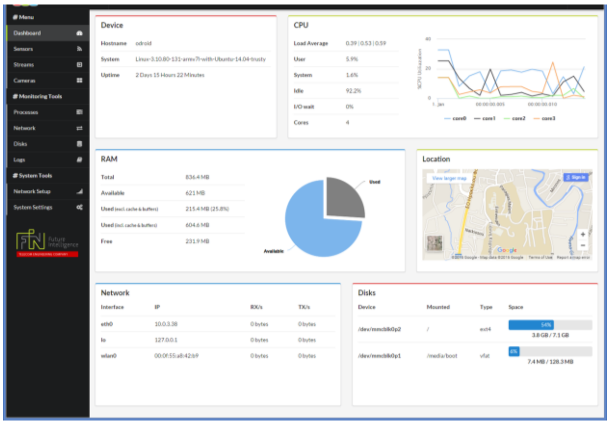

Fig. 4. Embedded system GUI

ESM: Embedded System Manager, controls and coordinates the other ES modules. Handles the communication between the ES and the RPU.

ESCOM: Embedded System Communicator, uploads Clips to the RPU, initiates VPN connection.

CG: Clip Generator, encapsulates Clip Data, Clip metadata (Clip Object) and ES Analytics Results in one Clip.

SM: Sensors Manager, acquires Sensors' data and feeds them to the Analytics, besides clipping them exploiting one of the available encoding schemes. It supports real time streaming upon request.

Analytics Module: Responsible of hosting different analytics algorithms downloaded from RPU.

The embedded system is a crucial component of the overall system since its handle the actual sensors and the 1st level of analysis. Finally, the embedded system in order to better managed and configured from the end user has its own GUI (Graphical user interface).

IV. Fog Node

The Fog Node consists of the following modules/components:

Video Analytics: This module can run more complex video analytics to provide a second verification of an identified security threat at the ES level.

Audio Analytics: This module can run more complex video analytics to provide a second verification of an identified security threat at the ES level.

Versatile Media Content Management System (VMCMS): The VMCMS is an ADITESS Ltd. solution for the efficient storage, management archiving, processing and logging of multimedia/heterogeneous content through a modular architecture. The platform is designed to fit the needs of a wide range of users through secure, yet, easy to deploy interfaces. VMCMS is characterized for its highly scalable, flexible and customizable design, which minimizes the deployment time. Fig. 5 is a screen shot of the VMCS GUI.

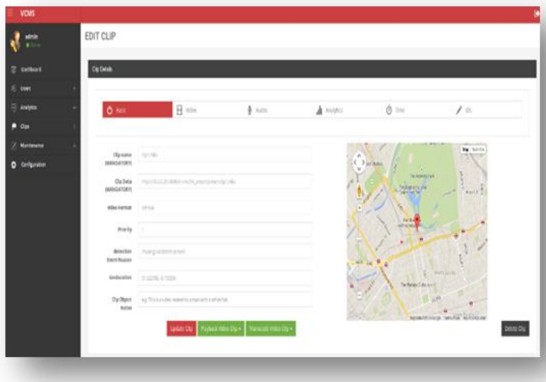

Fig. 5. ADITESS Ltd. graphical user interface of video content management.

Communicator (COM): It is designed to receive the Clips from the Embedded Systems, de-encapsulate, forward the contained Clip Object to VMCMS. It will handle VPN connections directly between the RPU and the ESs or from any intermediate equipment (aggregation of ESs at the edge e.g. building).

V. Testing

This section presents the performance evaluation of different Embedded Systems (ES) as an important part of the operation in the edge and before the fog node retrieves information for further analysis. The evaluation testbed consists of three different embedded systems, i) a single core ES1, ii) a dual core ES2 and iii) an octal core ES3.

Specific KPIs have been set for the performance evaluation:

- Clip data creation delay (CDCD). Measures the time needed to create a Clip Data from the incoming sensor's data stream (i.e. video camera) The observed time will be dependant to the Clip Data duration (how many seconds of scenery is contained in the Clip Data). the codec used to encode the data and of course the amount of processing load cause by other component processes of the embedded system.

- Operational Pipeline Delay (OPD). It is a composite metric and provides an estimation of the time needed to provide a Clip to the fog node. The observed time will be dependant to the codec used to encode the Clip Data, to the amount of processing load caused by the other components and processes of the Embedded System and to the available network bandwidth.

A. *Experimental Scenario*

Towards evaluating the performance of the three Embedded System prototypes, we executed in each one of the prototypes the same experimental scenario where: a) the Sensors Manager was streaming data, from a multimedia file (~20 minutes input dataset), to dummy video and audio analytics, b) the Clip Generator was generating Clips for every newly created Clip Data and c) The ES Communicator was uploading each newly generated Clip to the fog node. To obtain a better understanding of how the video encoding scheme in place affects the performance of the Embedded Systems, we executed the scenario in three variations where the utilised encoding schemes were MJPEG, MPEG4 and H264 respectively (TABLE II) for the average Clip Data duration per encoding scheme). It is noted here that the ADSL's 1 Mbps uplink has been used and therefore it was selected for emulating one of the lowest cost network solutions found. Finally, the configuration parameters for the used encoders (jpeg, mpeg4, h264 are presented in TABLE III—the encoding rate (bitrate) parameter was calculated based on Kush gauge formula [10]:

- pixel count x frame rate x motion factor x 0.07 ÷ 1000 = bit rate in kbps (pixel count=frame width x height, motion factor = {1, 2, 4})

The suggested encoding bit rate was ~ 1Mbps.

TABLE II. CLIP DATA AVERAGE DURATION

	ES1 (A10)	ES2 (A20)	ES3 - (XU3)
MJPEG	30 sec	30 sec	30 sec
MPEG4	32 sec	33 sec	32 sec
H264	32 sec	31 sec	31 sec

TABLE III. ENCODERS CONFIGURATION PAREMETERS

Encoder	Configuration parameters
MJPEG	Default parameters
MPEG4	bitrate=1000000, all rest were the default parameters
H264	bitrate=1000, tune=4, speed-preset=2, byte-stream=true, all rest were the default parameters

B. *Clip Data Creation Delay*

Concerning the Clip Data creation, Fig. 6 shows the overall delay (Clip Data Creation Delay-CDCD) per encoding scheme and Embedded System boards. It is clear that the h264 encoding scheme experiences a higher delay compared to the other two encoding schemes. To gain a better understanding of how each encoding scheme contributes to the measured CDCD, we subtract from it the corresponding Clip Data duration (see TABLE II). The results are depicted in Fig.6, where we can observe that, for all ES boards, h264 introduces the highest delay when compared to the other two encoding schemes. Considering these results, it looks more promising the use of mpeg4 or jpeg encoding schemes for the Clip Data Creation operation.

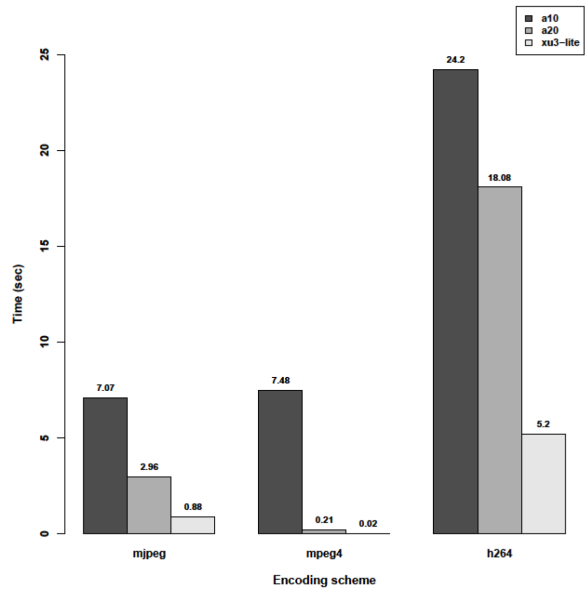

Fig. 6. Temporal overhead per encoding scheme

C. *Operational Pipeline*

Fig.7 presents the Operational Pipeline Delay (OPD) results per encoding scheme and for all the ES boards. The results indicate that the mpeg4 encoding scheme is the most efficient one. Considering that the higher contribution to OPD is the Clip uploading delay, an increase in the network bandwidth will reduce further the overall delay, thus improving the response time of the whole smart surveillance system including the fog nodes.

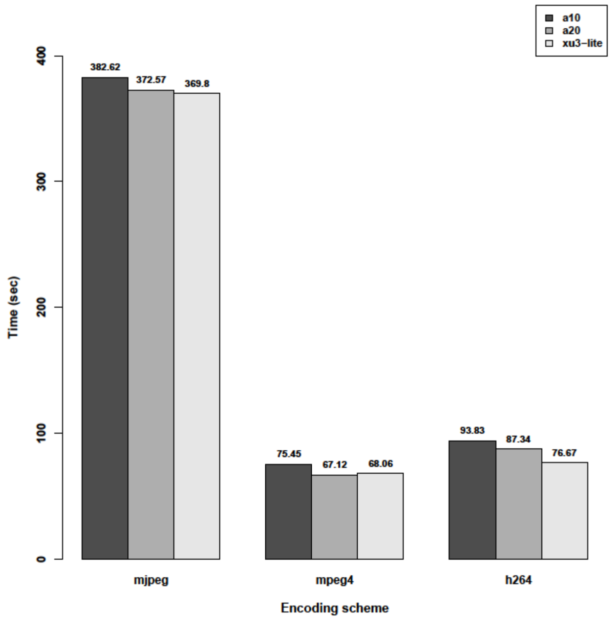

Fig. 7. Encoding scheme.

VI. Conclusions and Future Work

We presented in this paper a novel architecture for future surveillance systems based on edge and fog computing concept. We defined the architectural and software components in three different processing layers named i) the embedded system, ii) the fog node, and iii) the cloud. Three different versions of the edge node were presented with different storage, RAM and processing power. The final choice of the embedded system flavor is based on the end-user need and the application area. For the fog node, a Video Content Management Software was developed for processing and logging of multimedia/heterogeneous content. VCMS was based on a modular architecture and more functionalities will be planned. Lab tests with different encoding schemes including MJPEG, MPEG4 and H264 were conducted to validate both the efficiency of the embedded system and to investigate which one of the encoding schemes was more efficient. Lab results showed that mpeg4 encoding scheme was the most efficient.

In our future work, we plan to continue the trials for the whole system under a real environment where the performance of all different layers from the edge to fog and the cloud back-end will be evaluated with a number of test cases.

Acknowledgment

This work is partly funded by the European Union's Horizon 2020 research and innovation program under grant agreement No 732339. Project PrEstoCloud - Proactive Cloud Resources Management at the Edge for Efficient Real-Time Big Data Processing (http://prestocloud-project.eu).

References

[1] Burns, E., "Hadoop still too slow for real-time analysis applications" In TeckTarker. Available online at: http://searchbusinessanalytics.techtarget.com/feature/Hadoop-still-too-slow-for-real-time-analysis-applications, 2013.

[2] Cuomo, G., Martin, B.,K., Smith, K.,B., Ims, S., Rehn, H., Haberkorn, M., Parikh, J. (2013). Developing Edge Computing Applications. IBM White Paper, Avaialable online at: http://www.ibm.com/developerworks/cn/websphere/download/pdf/OnDemandEdgeComputing.pdf

[3] Cisco White Paper: Available online at: http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf, 2015.

[4] Mims, M.,. "Forget 'the Cloud'; 'the Fog' Is Tech's Future". In The Wall Street Journal. Available online at: http://www.wsj.com/articles/SB10001424052702304908304579566662320279406, 2014.

[5] IDC (2014). FutureScape: Worldwide Internet of Things 2015 Predictions. In International Data Corporation web conference. Available online at: https://www.idc.com/getdoc.jsp?containerId=prUS25291514

[6] Cisco (2013) RFP-2013-078. Fog Computing, Ecosystem, Architecture and Applications. Available online at: http://research.cisco.com/research

[7] Perry, T., (2013)What Comes After the Cloud? How About the Fog? In IEEE Spectrum. Available online at: http://spectrum.ieee.org/tech-talk/computing/networks/what-comes-after-the-cloud-how-about-the-fog

[8] Gatto, M., E., (2014) New Solutions on the Horizon—"Fog" or "Edge" Computing? In the National Law Review. Available online at: http://www.natlawreview.com/article/new-solutions-horizon-fog-or-edge-computing

[9] A20-OLinuXIno-LIME2 "Open Source Hardware Embedded ARM Linux single board computer with Allwinner A20 dual core CORTEX-A7 1GB RAM and Gigabit Ethernet", website available on-line at: https://www.olimex.com/Products/OLinuXino/A20/A20-OLinuXIno-LIME2/open-source-hardware

[10] ODROID-XU3 Lite "ODROID, Heterogeneous Multi-Processing (HMP) solution", website available on-line at: http://www.hardkernel.com/main/products/prdtinfo.php?gcode=G141351880955.

[11] Kush Amerasinghe, "H.264 for the rest of us", Available online: http://wwwimages.adobe.com/content/dam/Adobe/en/devnet/video/articles/h264_primer/h264_primer.pdf.