HAL
open science

# An Optimal Allocation of Tracking Tasks in a SW/HW Fog/Cloud Based Distributed Video Surveillance System

Ugo Sbai, Samy Meftali, Djamel Aouali

## To cite this version:

Ugo Sbai, Samy Meftali, Djamel Aouali. An Optimal Allocation of Tracking Tasks in a SW/HW Fog/Cloud Based Distributed Video Surveillance System. Journal of Communications, In press. hal-01716521

**HAL Id: hal-01716521**

**https://hal.science/hal-01716521v1**

Submitted on 23 Feb 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# An Optimal Allocation of Tracking Tasks in a SW/HW Fog/Cloud Based Distributed Video Surveillance System

Ugo Sbai (Univ. Lille France)
Ugo.sbai@univ-lille1.fr

Samy Meftali (Univ. Lille France)
Samy.meftali@univ-lille1.fr

Djamel Aouali (Dcarte Engineering)
D*jamel.aouali@dcarte.fr*

CRIStAL Laboratory. University of Lille 1. Batiment M3 extension. Avenue Carl Gauss.
59655 Villeneuve d'Ascq Cedex. FRANCE

***Abstract*—In this paper we propose a novel and systematic approach for *dynamic allocation of tasks in a video surveillance system using smart cameras and based on Cloud/Fog architecture. Tracking tasks arrive in the system in a random way and must be assigned to the available devices (cameras, Fog nodes and the Cloud). Our approach guarantees the best solution optimizing power consumption and communication cost over the system. The proposed methods uses an integer programming model and its effectiveness is shown on an application example.***

***Index Terms*—Smart camera, Power consumption, processing time, tracking, task assignment, Fog, Cloud, FPGA.**

## INTRODUCTION

Recently, many companies have adopted IoT (Internet of Things) applications. These applications basically use connected objects that are able to change their behavior, anticipate an action, and connect to other objects to perform an action.

These IoT services or applications, usually with sensors, produce a lot of important information to manage and control. They require an advanced hardware configurations to meet their needs in terms of computing power and storage.

However, these requirements are becoming increasingly difficult to satisfy in the case of multimedia applications such as video surveillance one in smart environment, for instance, such as buildings or cities is a recent trend [1]. In this context, the cloud computing is the most appropriate technical solution to meet the processing and computing needs of these IoT applications. At this level, the IoT-Cloud platform concept is reshaping the service delivery process. This concept executes a logic of centralized control and processing of data collected by physical sensors in a targeted area. In addition, the designers of the "IoT-Cloud" platforms are making considerable efforts to:

- Overcome the architectural design challenges of IoT-Cloud platforms.
- Simplify processing and analysis of data generated or collected in real time.
- Improve interoperability of equipment in these IoT-Cloud platforms to avoid errors and optimize service productivity.
- Improve the Qos of the different services launched on these platforms

On the other hand, the emergence of multimedia services and applications in these platforms (IoT-Cloud) considerably degrades the performance of the network access infrastructure. This prompted network solution designers to explore alternative paths to meet the demands of these services and applications. In other words, the exploration of network infrastructures capable of providing a better connectivity and availability of services.

The concept of "Fog computing" is proposed as a solution that can improve the performance of the IoT-Cloud infrastructure. This new FCIoT (FoG-Cloud-IoT) concept is defined as a distributed infrastructure in which a part of the service is managed and processed by smart components at a « network edge » in the network. This decentralized (distributed) approach allows due to these components (intermediate between the cloud and end-users) to relieve the access network and reduce the latency of the exchanges of these applications.

More recently, hardware components, called Field Programmable Gate Array (FPGAs), are taking amplitude in FCIoT (FoG-Cloud-IoT) platform architectures. These component are characterized by dynamic and partial configuration modes, allowing FCIoT platforms to quickly adapt to the change that has occurred during an event, to open up new issues for designers to improve the availability and continuity of services launched on these platforms. Such platforms represent, indeed, a serious challenge to overcome, especially in terms of deployment and positioning of FoGs.

On of the modern and most important challenges is a such context is how evolve an existing cloud based video surveillance system with multiple heterogeneous smart cameras and adapt it to a fog/cloud architecture in order to improve performances. I fact, current video surveillance systems contain a certain number of smart cameras having different computation powers and different available amounts of energy. Thus, when a company decide to evolve such a system, they generally add some fog elements and eventually new smart cameras that might be either software (SW) of hardware (HW) without removing the existing ones in order to maintain a reasonable cost.

In such a system, tracking tasks can be executed on any one of the available computation elements in the system, but their assignments might have significant impact on the performances of the whole system.

The objective of this work is to propose a novel approach allowing a dynamic, optimal and automatic assignment of hw/sw tracking tasks on the processing elements of a fog/cloud video surveillance system.

This paper is organized as follows: in Section 2, we give a review of several previous works fog/cloud based systems . In Section 3, we present our flexible and distributed video surveillance system. Section 4 describes the dynamic SW/HW tasks tracking assignment on the available computation elements over the fog/cloud based system. The proposed methodology is detailed on an application in Section 5. We conclude this paper and give some of our futur work in Section 6.

## RELATED WORK

Fog Computing is an application and hardware layer complementary to existing cloud solutions. The architecture of Fog computing makes it possible to meet the requirements of applications requiring real-time processing or massive processing of data collected by connected objects.

This king of architecture is a decentralized infrastructure in which data, computation, storage and applications are efficiently placed between the data source (sensors) and the cloud. In order to optimize the response times of IoT applications, FoG computing extends the services of cloud computing to the periphery of the network for the processing of collected data. These decentralized processing equipment may be routers, gateways or mobile devices. By reducing the path between connected objects and collected data processing equipment, Fog computing brings additional capacity to the Data Center, alleviates the load on the network and improves the latency of exchanges.

Many research works have focused on proposing approaches of video surveillance systems using smart cameras. For example, [1] describes the design and the implementation of an FPGA-based smart camera system for automated video surveillance. The complete system is prototyped on Xilinx ML510 FPGA platform and meets the real-time requirements of video surveillance applications while aiming at FPGA resource reduction. In [9], the authors present a dynamic task allocation method for smart cameras targeting traffic surveillance in distributed embedded systems. The allocation problem is formulated as a distributed constraint satisfaction problem (DCSP) and a distributed method for finding feasible allocations is presented. Finally, a cost function is used to determine the optimal allocation of tasks. In [7], the problem of change detection in the design of automated video surveillance system is addressed; appropriate selection of frames of significant changes can minimize the communication and processing overheads for such systems. The authors present the design of aVLSI architecture for change detection in a video sequence and its implementation on Virtex-IIPro FPGA platform; clustering-based scheme is used for change detection. The authors of [10] address the problem of managing power consumption in Smart Camera Networks (SCN) in order to extend the SCN lifetime; they propose a market-based solution where cameras bid for tasks using an adaptive utility function. They demonstrate that by assigning vision tasks to cameras in an energy-aware manner, it becomes possible to extend the network lifetime. The work presented in [14] provides a top level description of a complete automated video surveillance system along with the elaboration of different challenges/issues involved in its design and implementation. The work includes a comparative analysis of design methodologies and existing FPGA platforms and a complete design flow for prototyping the FPGA-based automated video surveillance system.

## DISTRIBUTED VIDEO SURVEILLANCE SYSTEM

The application domain that we target in this work is distributed video surveillance using heterogeneous smart cameras. At anytime each target is assigned to one camera, thus a SW or a HW tracking task is implemented and executed on this later. Figure 1.a illustrates a system composed by three cameras where each one tracks two targets.

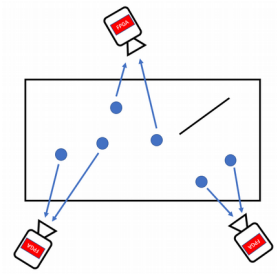


**Figure 1.a.** Six targets tracking

This system is highly dynamic in the sens that targets might move and new targets can appear at any moment.

Figure 1.b shows the case where a target moves behind an obstacle and leaves the field of view of the camera that was tracking it. It is therefore necessary to choose a new camera, which contains this target in its field of vision, in order to implement a corresponding tracking task. In the example of figure 2.b, there is a choice between two cameras.

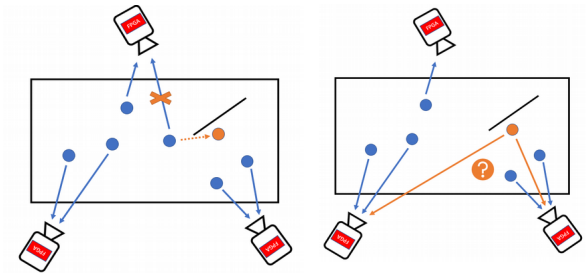

**Figure 1.b.** Six targets tracking

*A. System Architecture*

The system of video surveillance is based on a Cloud/Fog approach; it is composed by a set of smart cameras (devices, see Figure 2); some of them contain a dynamically reconfigurable FPGA (able to implement HW tasks), others ones have simply a processor and/or a GPU and thus can only implement SW tasks. These cameras communicate with Fog nodes.

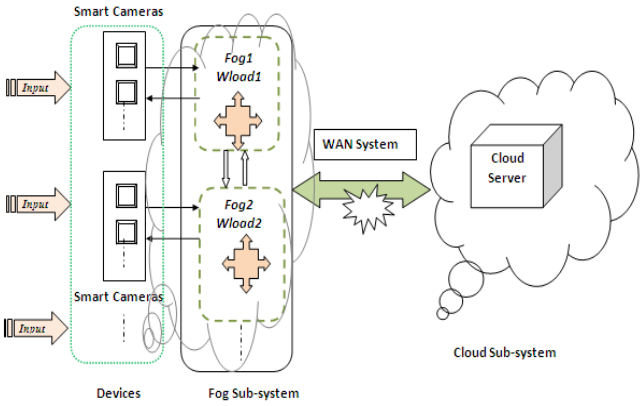


**Figure 2.** *General architecture of the system*

A Fog connects a certain number of cameras (depending on the strategy of placement of cameras and the covered area). Each Fog element contains a FPGA of medium size in terms of hardware resources (LUTs, DSPs, BRAMs, etc. as shown in figure 2).

These Fog nodes can communicate with each other (for negotiation and notification) and with the Cloud server which is composed of several big size FPGAs and/or soft-cores having a great computing power.

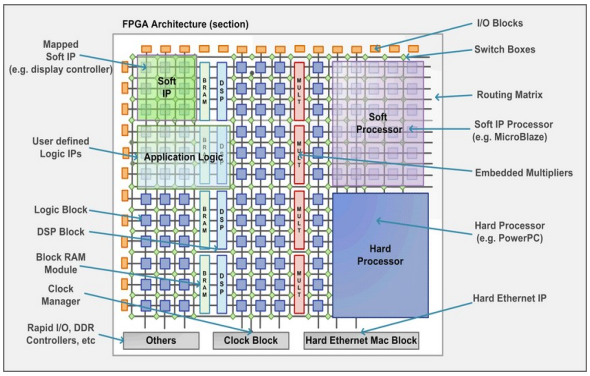


**Figure 3.** FPGA structure

*B. How does the system works?*

When a new target (for example, an object or a person that should be tracked) arrives in the system, any camera that detects it will inform its Fog by sending a message. Fogs send notification messages to the cloud containing new targets identification. The fog execute then an algorithm allowing the assignment of the new targets on the computation elements. In this tracking application, the process of tracking is executed on the cameras, on the Fog or also on the Cloud (at last resort) if no other choice.

*- Fog layer*

- On the Fog (and the cameras), tracking tasks are implemented on processors as SW tasks or on FPGAs as HW accelerators.
- The Fog nodes receive feeds from smart cameras (IoT devices) in real time.
- The Fog nodes run IoT applications in real time control with very quick response time (milliseconds)
- Fog nodes provide data storage (transient) and secure transportation
- The communication between a fog element and cameras is much faster than fog to cloud communication.

*- At the Cloud level*

- The Cloud server performs some tracking tasks (when needed) and executes the assignment algorithm.
- The Cloud server provide data storage and secure transportation

## MODEL FORMULATION

### A. *Objective function*

The objective is to assign dynamically new targets on the SW/ HW available elements on the system while optimizing criteria that have the greatest impact on system performances. In this work we selected two important criteria that are energy consumption and communication cost (time). Thus, this leads to a multi-criteria objective function.

### B. *Constraints*

The video surveillance system works under a set of constraints as follows:

- Each processing element (camera, fog or cloud) has a limited amount of energy. Obviously this available energy can be significantly different from an element to another depending on whether the element is connected to a power source or uses on battery. The execution of a tracking task consumes a certain amount of energy, thus the consumption corresponding to all tasks executed by any processing element must be lower than the available amount.
- At each device (camera) and the Fog nodes, the number of resources is limited; resources are assimilated to regions on FPGAs (zones) or to processes on sorfcores. Each region can implement a single task.
- Resources at the cloud are "theoretically unlimited"

### C. *Controllable variables*

$$X_{ij} = \left\{ \begin{matrix} 1 \; if \; target \; i \; is \; assigned \; to \; device \; (camera \; fog \; or \; cloud) \; j \\ O \; otherwise \end{matrix} \right\}$$

$\forall \; i \in \; 0..a \; and \; \forall \; j \in \; 0..n+m$

### D. Non-controllable variables

- $Com_i$: is the amount of the received data by the tracking task of the target *i*.

- $T_{ij}$: is the communication time (by data unit) from tracker of the target *i* and the device *j*.

$T_{ij}$ is set to :

- 0 for all indexes *j* corresponding to cameras able to track the target *i* and to infinity for all other cameras indexes.
- a big value **v** for all indexes *j* corresponding to fogs linked to cameras able to track the target *i* and to infinity for all other fog indexes.
- a very big value for *j* = 0 (index corresponding to the cloud).

- $P_{ij}$: is the amount of energy consumed by the device *j* to execute tracker *i*. To set this value we chose to use time slots of 8 hours corresponding generally to the dailly working time. Thus a target will be affected to a given device only if this later has enough energy to execute is (in addition to all its initial tasks) for 8 hours.

### E. *Integer linear programming model*

Even the problem we want to address is NP complete, it can be modeled using many different techniques such as heuristics, game theory, dynamic programming, mathematical programming, ..etc. However, we chose ILP because of its optimal nature and resolution tools availability.

We obtained the following objective function:

$$Min \; \alpha \sum_{i=0}^{a} Com_i \sum_{i=0}^{a} T_{ij} X_{ij} + \beta \sum_{i=0}^{a} \sum_{j=0}^{n+m} P_{ij} X_{ij}$$

where α and β are to constant values to be fixed by the systems designer. More α (resp. β) is bigger more the weight of communication cost (resp. power consumption) is increased in the objective function.

Under the constraints:

$$\sum_{j=0}^{n+m} X_{ij} = 1 \quad \forall \; i \in 0..a$$

$to \; ensure \; that \; each \; target \; will \; be \; assigned \; to \; one \; single \; device.$

$$\sum_{i=0}^{a} C_{ij} X_{ij} \leqslant CP_j \quad \forall \; j \in 1..n+m$$

$to \; ensure \; that \; each \; device \; (except \; the \; cloud) \; will \; not \; exceed$
$its \; capacity \; in \; terms \; of \; computation \; power.$

$$\sum_{i=0}^{a} P_{ii} X_{ij} \leqslant E_j \quad \forall \; j \in 1..n+m$$

$to \; ensure \; that \; each \; device \; (except \; the \; cloud) \; will \; not \; exceed$
$its \; capacity \; in \; terms \; of \; available \; energy.$

$X_{ij} \in 0..1 \quad \forall \; i \in 0..a \; and \; j \in 0..n+m$
$variables \; are \; booleans.$

## MODEL ANALYSIS AND COMPLEXITY

For a problem instance we obtain:

- at most NB_variables = M * (m + n + 1) variables.

Where M is the maximum new targets likely to apear in the system at any moment.

In realistic applications M could not be big and still around 2 or 3 new targets. Thus the number of variables might be in almost all practical cases lower than 3 (m + n + 1).

- NB_constraints = (a + 1) + 2*(n+ m) constraints.

The modeling of the tracking tasks assignment problem by the proposed integer linear programming approach presents some major advantages as:

- it is an exact method, which contrary to the heuristic based methods, gives an optimal solution,
- it is a very generic model which allows the integration of sensors and computing devices at all the layers of the systems architecture (HW/SW smart cameras, fog and cloud). It is also able to manage any number of targets,
- there are many available tools which permit the resolution of such a model. Cplex library is the one we used in our environment,
- the complexity for the proposed integer linear programming model remains very acceptable because the C code using Cplex library for the problem resolution is executed in the Cloud, using only few of the available, theoretically unlimited, resources.

## EXPERIMENTS

In order to illustrate effectiveness and efficiency of the proposed tracking tasks assignment method, we modeled and resolved several instances of the problem. The Cplex library has been used for modeling and resolution.

In these experiments we used a HP SERVER (Proliant DL980 G7) with 8 Xeon Six Core E6540 and 256 Giga bytes installed memory and 64 T bytes hard disc as a cloud layer. We used a benchmark close to realistic problems with one cloud server, 4 fogs and 20 cameras.

As this work focuses only on the assignment method which is executed on the cloud, we do not implement fog and cameras layers.

Thus after the resolution of the initial problem we added successively 2, 4, 5, 8, 10 and 12 new targets and resolved the tasks assignment problem after each modification of the initial configuration. All the solutions have been obtained in a very fast way (real-time) and took less then hundredth of a second. Such results confirms that the proposed models constitutes an efficient solution for middle size problem instances that we find in realistic video surveillance systems.

## CONCLUSION AND FUTURE WORK

In this paper, we presented an automatic a novel and automatic method allowing an optimal resolution of dynamic task assignment problem in Fog/Cloud video surveillance systems. We use an exact method to resolve the problem for the fixed criteria (total energy consumption and execution time). The proposed approach consists in an ILP model executed as a specific task on the cloud server, where it takes advantage from the important computation power available. This methodology gave interesting and efficient results, in terms of execution time, for various realistic instances.

Since some variables in our model are boolean, the problem is NP hard and then its resolution step can be slow depending on the number of such variables, especially if we want to execute the resolution task on an other device than the cloud. So, for the applications integrating lot of sensors and fog components, and we want more flexibility regarding the resolution task, we recommend the use of stochastic or heuristic methods instead of the linear model. Our future works will consist of exploring such solutions.

## REFERENCES


[1] Sanjay Singh, Sumeet Saurav, Ravi Saini, Atanendu S. Mandal, Santanu Chaudhury, (2017), FPGA-based Smart Camera System for Real-time Automated Video Surveillance, Communications in Computer and Information Science book series (CCIS, volume 711).

[2] SanMiguel, J. C., Micheloni, C., Shoop, K., Foresti, G. L., & Cavallaro, A. (2014). Self-reconfigurable smart camera networks. *Computer*, *47*(5), 67-73.

[3] OpenFog Consortium Architecture Working Group. (2017). OpenFog Reference Architecture for Fog Computing. *OPFRA001*, *20817*, 162.

[4] Osanaiye, O., Chen, S., Yan, Z., Lu, R., Choo, K., & Dlodlo, M. (2017). From cloud to fog computing: A review and a conceptual live VM migration framework. *IEEE Access*.

[5] M. Aazam and E.-N. Huh, (2014), Fog computing and smart gateway based communication for Cloud of Things," in Proc. IEEE Int. Conf. Future Internet Things Cloud (FiCloud), Barcelona, Spain, Aug. 2014, pp. 464-470.

[6] - S. Yi, C. Li, and Q. Li, ``A survey of fog computing: Concepts, applications and issues," in Proc. ACM Workshop Mobile Big Data, Hangzhou, China, 2015, pp. 37-42.

[7] Singh, S., Mandal, A. S., Shekhar, C., & Vohra, A. (2013). Real-Time implementation of Change Detection for Automated Video Surveillance System. *ISRN Electronics*, *2013*.

[8] M. Díaz, C. Martín, and B. Rubio, (2016), State-of-the-art, challenges, and open issues in the integration of Internet of Things and cloud computing," J. Netw. Comput. Appl., vol. 67, 99-117.

[9] Bramberger, M., Rinner, B., & Schwabach, H. (2005, June). Resource-aware dynamic task-allocation in clusters of embedded smart cameras by mobile agents. In *Proceedings of the IEE Intenational Workshop on Intelligent Environments*.

[10] Kyrkou, C., Laoudias, C., Theocharides, T., Panayiotou, C. G., & Polycarpou, M. (2016). Adaptive

energy-oriented multitask allocation in smart camera networks. *IEEE Embedded Systems Letters*, *8*(2), 37-40.

[11] Deng, R., Lu, R., Lai, C., Luan, T. H., & Liang, H. (2016). Optimal workload allocation in Fog-Cloud computing toward balanced delay and power consumption. *IEEE Internet of Things Journal*, *3*(6), 1171-1181.

[12] Dieber, B.; Micheloni, C. and Rinner, B., (2011), "Resource-aware coverage and task assignment in visual sensor networks", IEEE Trans. on Circuits and Systems for Video Technology, 21(10):1424-1437.

[13] Tessens, L.; Morbée, M.; Aghajan, H. and Philips, W., (2014) "Camera selection for tracking in distributed smart camera networks", ACM Trans. on Sensor Networks, 10(2): 1-33.

[14] Singh, S., Saurav, S., Shekhar, C., & Vohra, A. (2016). Prototyping an Automated Video Surveillance System Using FPGAs. *International Journal of Image, Graphics and Signal Processing*, *8*(8), 37.