OPERA

LOw Power Heterogeneous Architecture
for NExt Generation of SmaRt Infrastructure and Platforms
in Industrial and Societal Applications

Lesson Learned and Track Changes 3

Co-funded by the Horizon 2020
Framework Programme of the European Union

DELIVERABLE NUMBER	D2.2
DELIVERABLE TITLE	Lesson Learned and Track Changes 3
RESPONSIBLE AUTHOR	CSI Piemonte

GRANT AGREEMENT N.	688386
PROJECT REF. NO	H2020- 688386
PROJECT ACRONYM	OPERA
PROJECT FULL NAME	LOw Power Heterogeneous Architecture for Next Generation of SmaRt Infrastructure and Platform in Industrial and Societal Applications
STARTING DATE (DUR.)	01/01/2015
ENDING DATE	30/11/2018
PROJECT WEBSITE	www.operaproject.eu

WORKPACKAGE N. | TITLE	WP2 | Low Power Computing Requirements and Innovation Engineering
WORKPACKAGE LEADER	ISMB
DELIVERABLE N. | TITLE	D2.2 | Lesson Learned and Track Changes 3
RESPONSIBLE AUTHOR	Luca Scanavino - CSI
DATE OF DELIVERY (CONTRACTUAL)	M36
DATE OF DELIVERY (RE-SUBMITTED)	15/02/2019
VERSION | STATUS	V2.0|Final Version
NATURE	R(Report)
DISSEMINATION LEVEL	PU(Public)
AUTHORS (PARTNER)	Luca Scanavino (CSI)

VERSION	MODIFICATION(S)	DATE	AUTHOR(S)
0.1	Table of Contents Executive Summary Truck use case information VDI use case information	31/07/2018	Luca Scanavino - CSI
0.2	Traffic Monitoring use case information	23/10/2018	Jean-Christophe Maisonobe – LD38
0.3	Conforming LD38 contribution Updating of Truck use case information Updating of VDI use case information	31/10/2018	Luca Scanavino - CSI
0.4	IBM Internal review	05/11/2018	Joel Nider (IBM)
0.5	Nallatech Internal review	14/11/2018	Richard Chamberlain (Nallatech)
1.0	Final Version	14/11/2018	Luca Scanavino - CSI
1.1	In response to reviewer comments, we aligned D2.2 structure to D2.9 and D2.10	24/01/2019	Luca Scanavino – CSI
1.2	Updated version of the deliverable version 1.2	25/01/2019	Alberto Scionti (ISMB), Joel Nider (IBM)
1.3	Nallatech contribution	31/01/2019	Richard Chamberlain (Nallatech)
1.4	ST contribution	07/02/2019	Francesco Papariello (ST)
1.5	Added lessons learned from the virtual memory research	08/02/2019	Idan Yaniv (Technion)
1.6	LD38 contribution	11/02/2019	Jean-Christophe Maisonobe (LD38)
2.0	Final Version	14/02/2019	Luca Scanavino – CSI

PARTICIPANTS		CONTACT
STMICROELECTRONICS SRL		Giulio Urlini Email: Giulio.urlini@st.com
IBM ISRAEL SCIENCE AND TECHNOLOGY LTD		Joel Nider Email: joeln@il.ibm.com
HEWLETT PACKARD CENTRE DE COMPETENCES (FRANCE)		Gallig Renaud Email: gallig.renaud@hpe.com
NALLATECH LTD		Craig Petrie Email: c.petrie@molex.com
ISTITUTO SUPERIORE MARIO BOELLA		Olivier Terzo Email: terzo@ismb.it
TECHNION ISRAEL INSTITUTE OF TECHNOLOGY		Dan Tsafrir Email: dan@cs.technion.ac.il
CSI PIEMONTE		Vittorio Vallero Email: vittorio.vallero@csi.it
NEAVIA TECHNOLOGIES		Stéphane Gervais Email: s.gervais@lacroix.fr
CERIOS GREEN BV		Frank Verhagen Email: frank.verhagen@certios.nl
TESEO SPA		Stefano Serra Email: stefano.serra@eiffage.com
DEPARTEMENT DE L'ISERE		Olivier Latouille Email: olivier.latouille@isere.fr

ACRONYMS LIST

Acronym	Description
ANN	Approximate Nearest Neighbour
BNN	Binary Neural Network
BSP	Board Support Package
CAPI	Coherent Accelerator Processor Interface
ECRAE	Efficient Cloud Allocation Engine
FGPA	Field Programmable Gate Array
GPU	Graphics Processing Unit
HDLPS	High density low power Server
IPS	Integration POWER8 bases server and CAPI protocol
LL	Lesson Learned
LXC	Linux Containers –a specific implementation of the container technology
LXD	Linux Containers – an easier to use interface to manage containers
NIC	Network Interface Card
NUMA	Non-Uniform Memory Access
ODP	On-Demand Paging
PS	Protocols and Standards
PST	Platforms and software technologies
RDMA	Remote Direct Memory Access
RoCE	RDMA over Converged Ethernet
RWA	Reconfigurable wireless antennas
SoC	System-on-Chip
TLB	Translation Lookaside Buffer
UC	Use Case
ULP	Ultra-Low Power Computing System
WP	Work Package
YOLO	You Only Look Once

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Position of the Deliverable within OPERA Project

One of the main target of WP2, as reported in D2.9 and D2.10, is to capture, analyse and harmonise different lessons learned from technological WPs (WP3, WP4, WP5 and WP6) and define their impact on the three use cases (WP7). In this way, it's possible to find out the gaps, between expected results and the actual achieved ones. During the previous cycles, in case of gap we defined corrective actions to reduce it during the subsequent validation phases, instead in this deliverable we report the residual difference, the related reasons and the updated lesson learned.

Description of the Deliverable

The structure follows the same pattern used for D2.9 and D2.10, without repeating the methodology description and lesson learned organization already described in the previous deliverables, that means:
Section 1 provides a detailed insight of the Lesson Learned (LL) of each technological WP, considering expected results, as well as initial requirements and constraints.
Section 2 analyses the effective results obtained in the three use cases, highlighting any difference with the expected ones.
Section 3 lesson learned acquired for each use case.
Section 4 concludes this report with the final considerations.

TABLE OF CONTENTS

1 TECHNOLOGICAL WORK-PACKAGE LESSONS LEARNED

In this section we describe the lesson learned from each technological work package (WP3, WP4, WP5, and WP6), we refer to these WPs as 'technological' since they are aimed at developing the set of technologies (hardware and software) and integrated solutions that serve as a basis in the use cases.

In the following figure, we report the last architecture that provides a reference model for the use cases:

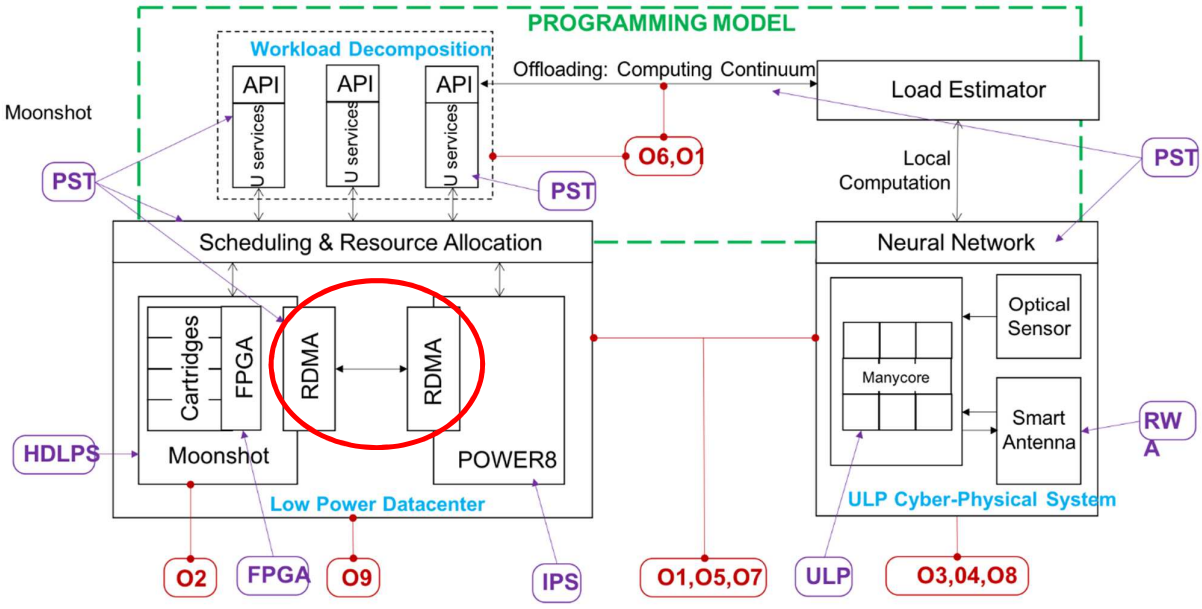

Figure 1 - Overall OPERA architecture highlighting the main addressed objectives.

According to the previous architecture, in the following table the description of each of the objectives is given:

OBJ	Description
01	HETEROGENEOUS ARCHITECTURE
02	NEW GENERATION SERVERS
03	ULTRA LOW POWER ARCHITECTURE
04	HIGH PERFORMANCE ON ULP ARCHITECTURE (SoC)
05	METRICS FOR ENERGY EFFICIENCY
06	WORKLOAD MANAGEMENT
07	INNOVATIVE POWER MANAGEMENT
08	ULP WORKING PROTOTYPE ON REAL TIME CONSTRAINT
09	SCALABLE SMALL FORM FACTOR DATACENTRE

Table 1 – Objects Architecture Description

Figure 1 highlights the main difference from the previous iteration: we substituted FPGA card with RDMA support as reported in WP5 deliverables.

It follows a description of the obtained results, highlighting the specific lesson learned and, where possible, to highlight the contribution to the implementation of the offloading mechanism. The LL incorporates the analysis of the outcomes of each task composing the WPs, as well as it takes into

consideration the outcomes reported in D2.10. Finally, if required, the causes of residual differences between outcomes and targets.

The previous figure (Figure 1) shows the main OPERA architecture with the main project's objectives highlighted. Such figure helps the reader to better understand how the various WPs and related tasks contributed to the achievement of the overall project objectives.

1.1 WP3 – ULTRA LOW POWER EMBEDDED ARCHITECTURE DEVELOPMENT

The work in WP3 is focused on the definition of hardware and software requirements of the components of an integrated system aimed at supporting local computer vision processing (as foreseen by Traffic Monitoring use case). This WP also aims at exploring state of the art technologies and techniques for hardware components integration, as well as for software development. Specifically, this WP focused on the integration of two ultra-low power (ULP) many-core System-on-Chips (SoCs) with an efficient radio communication subsystem:

- SecSoC platform
- Orlando platform, especially designed for implementing computer vision applications based on a state-of-the-art convolutional neural network (CNN), thanks to its Convolutional HW accelerators.

The tasks identified in WP3 are contributing on many of the project's objectives, while they have a focus on objectives O1 (Heterogeneous architecture), O3 (Ultra-low power architecture), O4 (High performance on ULP architecture), O6 (Workload management consolidation on scalable small form data center), and O8 (ULP working prototype on real time constraint for traffic monitoring) – see Figure 1 - Overall OPERA architecture highlighting the main addressed objectives. Finally, activities carried out in this WP also helped towards the design and implementation of the offloading mechanism, by integrating the required features that enable the application running on the ULP device to 'offload' heavy computational tasks on a remote low power server.

WP3 is composed by 4 tasks. **Task 3.1 – HW/SW requirements for new generation of ULP vision node –** aims at defining the software and hardware requirements for improving over the state-of-the-art systems. The results of these activities are available in deliverable D3.1. **Task 3.2 – Hardware design, development and testing –** aims at applying the (hardware) requirements and components defined in the Task 3.1 to a real platform, which is designed and tested. **Task 3.3 – Software functionality development and testing –** aims at developing, integrating and harmonizing the software components into the platform developed in Task T3.2. The designed system is tested along with the hardware platform. **Task 3.4 – RF module design, development and testing –** is devoted to design and implement an energy efficient radio frequency module.

1.1.1 Lesson learned

LESSON LEARNED WP3
General
WP3 is focused on the development of ultra-low power devices and their interconnection with remote servers. The main challenges of the third cycle were related to the use of the Orlando board, which is not mature and has no mature tools for software development compared to the SecSoC board. Instead, for what concerns the first test site, using the SecSoC board, some difficulties were encountered due to the prolonged exposure to adverse weather conditions.
Task 3.1
This task ended in M10.

Task 3.2
Issues on test site 1 during summer 2018, whose solution could be simplified by testing the full prototype in lab.
Task 3.3
Issues related to the Orlando platform and its tools, which are not mature.
Task 3.4
A full prototype installation in lab (including SecSoC/Orlando, ULP, Nucleo, antenna management board and antenna) could have helped debugging and validating the full system.

Table 2 - WP3 Lesson Learned

Task 3.1 – HW/SW requirements for new generation of ULP vision node

Activities for this task ended in M10.

Task 3.2 – Hardware design, development and testing

We experienced some issues on test site 1 during summer 2018 due to some electrical problems in one of the boards composing the ULP node, probably caused by the fact that all that circuitry is placed in an outdoor environment, basically exposed to the weather, even if protected by a plastic box. Detecting and fixing this issue required three physical interventions on site by Teseo, ST and ISMB and, until the problem was fixed, it was not possible for Neavia to remotely reconfigure the SecSoC, and this caused a delay in the development and integration of wrong way detection software.

Solution of this issue could have been simplified by testing the full prototype in lab immediately after the first intervention, instead the problem was underestimated during the first intervention and the test in lab was performed only after the second intervention.

Task 3.3 – Software functionality development and testing

On site 2, the use of the Orlando platform, which is a development board with no mature development toolchain and tools, caused a delay in the development and testing of cycle detection and counting use case. After some time spent trying to improve the Orlando development tools, Neavia and ST agreed on partitioning the software between Orlando (CNN + detection, done by ST) and Nucleo (tracking, counting and offloading, done by Neavia). Time could be saved if the decision to split software responsibility between ST and Neavia had been taken in advance.

Task 3.4 – RF module design, development and testing

A full prototype installation in lab (including SecSoC/Orlando, ULP, Nucleo, antenna management board and antenna) could have helped debugging and validating the full system before any intervention on site, preventing loss of time due to hardware and communication issues, and decreasing the impacts on software development, testing, debugging, evaluation and measurements. This was not completely performed because the nodes are very complex systems composed of many devices and some of them were not fully available (e.g. the antenna).

1.1.2 Achieved results

WP3 activity has been focused, for the reporting period, to integration of the various components of the ULP platform (that is at the basis of the edge node in the computing continuum envisioned OPERA architecture). Such integration regarded both hardware components (SecSoC, Orlando, Nucleo board, Raspberry, RF-antenna, etc.) and software elements. Many issues were encountered mainly due to the

partners using different developing tools and methodologies. To solve this issue, a common set of tools and sharing system has been adopted. Moreover, some issues were related to the toolchain of the Orlando board, which is not mature to be used outside of ST. This forced a redistribution of the software development effort between ST (developing the lower level layers of the software) and Neavia (developing the higher level layers of the software).

The traffic congestion detection application has been fully tested in different light and weather conditions, reaching a good level of maturity and a TRL7, "*system prototype demonstration in operational environment*".

Due to the issues encountered on test site 1 during summer 2018, the wrong way detection algorithm has been tested only in particular light conditions, limiting the operational range of the prototype.

The issues related to the development tools of the Orlando board, plus the burden inherently connected to the use of a new board (instead of the SecSoC) to implement the use cases of test site 2, introduced some delays and the cycle detection and counting application could be tested only for a limited period of time, and not continuously like the first installation site.

Even if the system has been tested completely in the operational environment, the TRL classification in this case is limited to TRL5 "technology validated in relevant environment".

In all these use cases, the ULP node is able to autonomously detect the different situations (traffic congestion, wrong way, presence of bicycles…), offloading data very unfrequently only in particular situations, hence dramatically reducing the power consumed by the communication link and by the remote servers.

1.2 WP4 – ENERGY EFFICIENCY ON HETEROGENEOUS ARCHITECTURE

This work package is focused on researching on models, methods and algorithms to be used to measure, prove and monitor the energy efficiency of developed OPERA solutions (both hardware and software). To define what model is suitable, the targets and criteria need to be defined during the completion of this process. The tasks identified in WP4 are contributing to most of the project's objectives and have a focus on objectives O1 (Heterogeneous architecture), O5 (Metrics for energy efficiency) and O7 (Innovative power management) – see Figure 1 . Specifically, **Task 4.1 – Research energy efficiency requirements –** is focused in researching alternative energy efficiency models and methods, as well as to analyse and evaluate, against the confirmed criteria, such models and methods, having in mind the concepts and products of OPERA. The finding of these criteria involves academic research, interviewing academics, research organizations, industry organizations and visiting the relevant seminars with energy efficiency as a main topic. **Task 4.2 – Testing alternative energy efficiency methods and models –** aims at providing a preliminary power analysis to identify system (both referring to the ULP-based platform and the platform based on low power servers) power consumption in different scenarios. Such preliminary analysis is based also on inputs provided by WP2. Also, large use of the results of the 'cluster collaboration' FP7-smartcities-2013 (ICT) Objective ICT-2013.6.2 has been of help in achieving task and WP objectives. Finally, energy harvesting techniques are also investigated and tested in different project scenarios. **Task 4.3 – Energy efficiency analysis –** aims at analysing the results of applying methodologies and models researched on T4.1 and T4.3, when use case context is taken into consideration. **Task 4.4 – Energy-aware server interconnect –** is devoted to minimizing latency between compute nodes. One of the investigated mechanisms was based on using the CAPI protocol (i.e., a cache-coherency protocol over PCIe), which allows to replace standard Ethernet links, thereby reducing latency and power(energy) consumption. To this end, in this task the use of fiber optics to connect the components at the physical layer was initially planned for investigation. However, as highlighted also in Figure 1, we moved towards a solution based on a hardware board supporting RDMA communication protocol.

1.2.1 Lesson learned

The lessons learned in WP4 are distinguished in general lessons learned and where possible, related to the different tasks. The LL is reported in the following (Table 3):

LESSON LEARNED WP4
General
In the last stretch of the project, cooperation between the various work packages was much improved. The regularity of the WP4 meetings combined with the attendance of the WP4.x leaders in the calls from other work packages paid off.
The online meetings have been integrated into our (bi-)weekly agendas, general attendance was good. In addition to the WP4 meetings, the persons responsible for WP4 tasks also joined in relevant calls for WP3, 5 and 7 which facilitated the transfer of relevant data needed for the creation of the WP4 deliverables.
Task 4.1
This task has been closed.
Task 4.2
Evaluating virtualized setups (i.e., applications running inside virtual machines) required automated software infrastructure. Due to time limits, we made do with a basic ssh-based communication between the host and the guest to pass instructions.
The first power estimation sized the energy harvesting system with a wireless data throughput of 1Mbps. However, due to an unexpected hardware limitation, the reliable throughput was only 50kbps, which increased the radio consumption (since the module will be active for more time than expected to transmit data). For this reason, energy harvesting component has been sized again and the days of autonomy reduced from 7 to 5.
Task 4.3
Focus needed to be put on obtaining hard data, all parties needed to contribute their respective market expertise in order to validate the market potential. Integration of WP2 and WP4 required to be monitored; activities in each of these work packages provided useful cross work package data.
Task 4.4
It was not possible to use FPGA boards for interfacing nodes and supporting CAPI. For this purpose, we revert in using off-the-shelf Mellanox ConnectX-5 board(s).
We needed to rethink the methodology used to measure the energy consumption, since the chosen board(s) do not natively integrate any hardware feature to get energy/power consumption data.

Table 3 - WP4 Lesson Learned

Task 4.1 – Research energy efficiency requirements

This activity has been already completed.

Task 4.2 – Testing alternative energy efficiency methods and models

This task is aimed at evaluating energy efficiency methods and models at both data center server side and remote ULP devices. The research efforts in the reported period were focused on two methods to improve the energy efficiency:

- Improving performance of servers running memory intensive benchmarks. Since 75% of x86 server workloads are running in the cloud, we were specifically interested in mitigating virtual memory overheads in cloud setups, i.e., when applications are hosted in virtual machines. Evaluating cloud setups was challenging because it required simulating and tracing virtual machines quickly and automatically. To this end, we developed a basic ssh-based communication between the host and the guest to pass instructions.
- Developing an energy harvesting system consisting of three modules: (1) photovoltaic panel that transforms solar radiation to electrical energy, (2) battery to store excess energy that will be used later when radiation is not enough, and (3) battery manager to control and optimize stored energy levels. The first power estimation reported in D4.5 sized the energy harvesting system with a wireless data throughput of 1Mbps (worst case for 802.11b/g wireless communication). However, due to an unexpected hardware limitation (explained in D3.4), we found a reliable throughput of 50kbps. This issue affects the radio consumption since the module will be active for more time than expected (i.e. to perform data transmission). For this reason, energy harvesting component has been sized again and the days of autonomy reduced from 7 to 5.

Task 4.3 – Energy efficiency analysis

From the last period, we got the following outcomes. The reason behind offloading bicycle counting from the camera to the remote low power server should ultimately be 'energy efficiency'. This efficiency would be lowered by the ability of the camera to analyse the pictures for bicycle counting because this would require CPU time and more memory in the camera. This is undesirable because it would lower the autonomy time of the camera.

From a broader viewpoint, all the partners should be more active in providing their inputs in terms of expertise for improving the exploitation of the market potential. To this end, a closed exchange of information, and main outcomes between WP2 and WP4 has been seen as useful.

Task 4.4 – Energy-aware server interconnect

Task 4.4 underwent major revision after losing support for the FPGA based design. Originally, we planned to use energy measurements coming from the FPGA by using built-in hardware for that purpose. Since we must now use off-the-shelf equipment, we were forced to find alternative methods for measuring energy consumption. The cards we chose (Mellanox ConnectX-5 [1]) do not have on-board energy measurement hardware. Therefore, we had to turn to external methods of energy measurement. At a high level, we wanted to measure energy at the server level, while it was performing a migration. Since we know the energy consumption of the NIC is a small percentage of the overall server consumption (~15W / 750 W = 2%) we did not expect to see much of a difference by measuring at that level. At a micro level, we can measure the energy consumption at the PCIe bus, since this is where the NIC gets all of its power from. On the positive side, we moved to use a standard protocol (RoCE) which means it was much more likely to find a place in a real deployment. In addition, practically all server vendors support PCIe which means it is much easier to adopt the Mellanox NIC solution in other architectures, than a pure CAPI solution.

1.2.2 Achieved results

During the reporting period, tasks' activities proceeded, and results have been recorded.

The Technion developed a new methodology for virtual memory research, combining micro-architectural simulation of memory-intensive benchmarks with energy models calibrated from real CPU measurements. The newly proposed hashed page table was evaluated using this methodology and was shown to be 6%--32% more effective compared to the radix page table design of existing x86-64 hardware.

The energy harvesting system has been tested in the field to check its proper functionality. The voltage generated by the solar panel has been measured (20V peak) and found to be correspondent to the datasheet. This preliminary verification allowed the tests performed on the overall system (i.e., in function

with Orlando camera board and wireless radio). The current power consumption allows for five days of autonomy.

On the energy efficiency of the ULP platform, the issues with the antenna coupling have been resolved and measurements have been taken both in field as well as in lab. A detailed set of data has been collected, showing the final results in energy efficiency improvements of the OPERA project. Regarding the energy aware interconnection of servers, the Mellanox ConnectX-5 board, allowed us to test energy efficient interconnection among heterogeneous servers, leveraging a more standardized RoCE protocol. This solution has been found to be enough to show usability with the OPERA solution for Linux containers.

1.3 WP5 – OPTIMIZED WORKLOAD MANAGEMENT ON HETEROGENEOUS ARCHITECTURE

WP5 focuses on researching mainly software solutions enabling the construction of (next-generation) power-aware data centers. Specifically, the activities carried out in this work package are aimed at demonstrating power consumption reduction (and consequently energy consumption reduction) by largely improving the efficiency of the underlying infrastructural resources. To this end, the WP researches methods and algorithms for scheduling tasks to most appropriate compute resources. Workloads are analyzed and characterized, as well as multiple Cloud computing models that allow software components to schedule tasks most effectively by exposing new functionalities in the Cloud management stack. WP5 activities contribute to most of the project's objectives, although they focus on: objectives O1 (Heterogeneous architecture), O2 (New generation low power servers), O6 (Workload management consolidation on scalable small form factor data center), and O9 (Scalable small form factor data center) – see Figure 1. Within the WP5 there are 5 tasks. **Task 5.1 – Workload characterization –** aims at characterizing Cloud workloads in terms of their impact on the physical resources (e.g., CPU and memory utilization). Such kind of analysis can be reflected in adapting an application descriptor (e.g., OASIS TOSCA) by specifying information on performance and resource requirements for a given application, as well as to drive scheduling and resource allocation algorithms; thus, becoming a common ground for designing innovative power management and placement policies and consolidation algorithms. Also, simulations can be performed in order to test the quality of the models. **Task 5.2 – Power-aware cloud model –** will investigate on the efficiency of different Cloud computing models. Specifically, a comparison between traditional models (e.g., applications hosted locally on a server or virtual machines) and Cloud services will be performed. Also, simulations can be performed in order to test the quality of the models. The goal of **Task 5.3** – **Power-aware cloud management –** is the research on, design and implementation of an interface between the software component devoted to allocating infrastructure resources and other modules composing a typical Cloud management stack (e.g., the various modules available in OpenStack). Such interface is exploited to drive the software component developed in T5.4 to reserve computing resources for virtual machines (VMs) and/or Containers, taking into account also power consumption profiles of the computing nodes. A prototype based on the OpenStack framework has been developed. **Task 5.4** – **Resource allocation –** specifically aims at designing and implementing a software component responsible for allocating data computational resources to the incoming jobs (VMs and Containers), as well as balancing the already allocated workload, to improve the energy(power) efficiency of the whole data center. Simulations are performed to assess the effectiveness of the selected allocation strategies. Finally, **Task 5.5** – **Profiling workloads –** surveys state-of-the-art techniques for estimating the power/energy consumption of a data center computing node. In addition, it also establishes a methodology to reduce the noise in energy measurements when multi-socket (NUMA configuration) processors are taken into account. This task applies selected energy measurement techniques both to standard benchmarks and use cases. Finally, in this task efficient memory management techniques will be researched too.

1.3.1 Lesson learned

The lessons learned in WP5 are separated in to general lessons learned and where possible, specific to a particular task. The lessons learned are reported in the following (Table 4):

LESSON LEARNED WP5
General
Beside the technical work done in this work package, one of the main lesson learned we got regards the organisation and structure of the tasks. Basically, we have seen for two tasks an overlapping in terms of responsibilities with other two tasks. Specifically, T5.1 and T5.3 where partially overlapped with T5.2 and T5.4 respectively. To this end, we refocused the activities on such tasks to better distribute the work and responsibilities, specifically between T5.1 and T5.2, and between T5.3 and T5.4. Also, technical activities have been carried out. Specifically, we successfully decomposed into micro services and deployed them in the CSI testbed infrastructure two applications selected in the context of VDI use case. However, given the immaturity of some of the technologies not foreseen at the beginning we delayed in completing this activity. A better preliminary analysis and plan will be of worth for future activities touching cutting-edge technologies. More effort has also spent in designing and implementing power-aware allocation and workload consolidation policies, since this represents a great step beyond the state-of-the-art. We also analysed the advantage of the proposed solutions, as well as how to speed up the execution of such algorithms.
Task 5.1
Analysis of the effort allocation in tasks T5.1 and T5.2 revealed, in some cases, an overlapping in terms of responsibilities.
Successful decomposition and deployment of the applications from the VDI use case on the CSI testbed for testing purposes.
Task 5.2
Technological challenges (immaturity of some of the technologies used) caused us to execute the task slower than originally planned. Also, interactions and dependencies between partners caused some delays. Once the overlap in task definitions was discovered, we should have pushed the coordinator harder to submit the amendment on time, which would have reduced uncertainty in the project.
Task 5.3
Overlapping responsibilities and effort allocation between tasks T5.3 and T5.4.
Interfacing with OpenStack module achieved by leveraging on TOSCA description format (exploiting a translation module that converts TOSCA to internal OpenStack directives for deployment).
Task 5.4
The greedy strategy for initial placement of application's components (i.e., containers or VMs) tested using a simulation environment (to assess the advantage over conventional allocation policies), and correctly integrated into the orchestration platform.
A workload consolidation strategy tested using a simulation environment. It is based on a published optimization model. For solving such model, we implemented a heuristic based on an Evolutionary Strategy (ES) algorithm (better suited for discrete optimization problems than PSO), which can take advantage from multi-core processors (also low power) available in data centers.
We also evaluated the opportunity to offload ES algorithm on accelerators (FPGA or GPU) for fast convergence of the searched solutions.

Task 5.5
Developing a flexible memory allocator (controlling which memory regions are backed with huge pages) is more challenging that it seemed at first.
Modifying and customizing 'glibc' library to control memory allocation flexibly with huge pages was more challenging than expected due to auto generated code and code complexity; there are several code parts in glibc that are auto generated and it's difficult to follow all of these code pieces and debugging the code during development process.
Intercepting memory allocation APIs is sensitive and could lead to either infinite recursive calls to memory allocation APIs or to intercept memory allocation calls partially and not all of them which causes to non-contiguous memory allocations.
Using LD_PRELOAD to load the memory allocator library has some bugs and known issues: there is an opened bug on loader code in glibc that libraries are initialized in reverse order of their appearance in search path when using LD_PRELOAD, and not by using some depth initialization like DFS.
Building performance models requires a lot of experimental data. We used manual process for collecting this data and leave the automation of this process for future work.

Table 4 - WP5 Lesson Learned

Task 5.1 – Workload characterization

Our biggest lesson from T5.1 is about the division of labour. To this end, we accomplished our original goals for this task by better aligning the efforts and objectives in a more logical fashion. There is a large overlap in responsibility between this task and T5.2, including the comparison of various Cloud models, and the deployment of test apps to measure performance of such models. There is also an overlap of responsibilities with T5.3, which aims to provide the container migration mechanism and OpenStack module (ECRAE) that are also listed as objectives of this task. As mentioned earlier, we reorganized the activities and the effort to meet these objectives. We have decomposed the applications selected within the VDI use case; they have been deployed for testing on the CSI testbed.

Task 5.2 – Power-aware cloud model

Through the experiments in the CSI testbed (see deliverable D7.7, D7.8 and D7.9), we have shown the advantages of shared SaaS applications for a workplace over a traditional VDI deployment. There were several challenges that caused us to execute the task slower than originally planned. Interactions and dependencies between partners caused some delays, as well as the immaturity of the underlying technology. One of observations that arises from working with cutting-edge technology is that the not adequate maturity level of such technologies makes difficult their integration within a production system. Indeed, missing features, poor integration between components and availability of patches can slow down the deployment phase.

Task 5.3 – Power-aware cloud management

Within the VDI use case, the first application to be ported on container-based SaaS model has been selected. The 'OwnCloud' has been successfully ported to the microservice model showing larger energy efficiency w.r.t., RDS model. We leveraged on this application to understand how OpenStack could be configured to correctly launch application containers. To this end, the OpenStack component selected for the installation into the CSI testbed were enough to drive correctly this activity. We also achieved the creation of the corresponding TOSCA descriptor, which has been tested (i.e., we verified that through this descriptor file was possible to launch correctly the service containers). The test has been done by passing the descriptor file to the ECRAE orchestrator module we developed. A second application (OpenXchange) has been successfully ported to the microservice (containerized) model. We also tested this application within the CSI testbed and using the orchestration solution we developed.

Task 5.4 – Resource allocation

This task mainly focused on the implementation and test of the designed power(energy) allocation policies (see the greedy allocation policy and workload consolidation reported in D5.8 and D5.9). Specifically, we selected the CloudSim Plus [2] environment to prove the effectiveness of greedy allocation strategy. Thus, we used CloudSim Plus environment to assess the capabilities of the designed greedy allocation strategy. We integrated such power(energy)-aware policy within the orchestrator module (namely ECRAE – see D5.8 and D5.9) and verified that it was able to correctly deploy microservices on the CSI testbed infrastructure. The orchestrator module leverages on the internal knowledge base (KB) to have an abstracted view of the status of the data center machines. Another activity carried out in this task has been the implementation of the heuristic used to optimize and consolidate the workload in the data center. To this end, we implemented and tested the ES algorithm in a simulation environment. The initial algorithm for performing such optimization phase was based on the PSO algorithm. However, initial tests showed poor performance and longer time to converge than expected. Thus, we revert to a different implementation based on an Evolutionary Strategy algorithm, which demonstrated more suited for the nature of the problem (that is not continuous). Furthermore, such algorithm can take advantage from the availability of multiple cores in modern CPUs.

Task 5.5 – Profiling workloads

As Technion is an academic partner and T5.5 is a research task, it was expected to have several challenges to address in this task, since it tries to explore new methods that was not used before. Specifically, one of the main focus of the activities carried out concerns the implementation of a more efficient memory management subsystem. To this end, part of the research activities has been devoted to integrating a memory allocator within the standard 'glibc' library, which should be able to manage also huge memory pages. Despite the initial plan foreseen, several technical challenges to solve arose, e.g., reported bugs of the 'glibc' standard library. A large effort has been to complete the integration of the new memory allocator and thus demonstrate superiority of the proposed solution over the current one in use.

1.3.2 Achieved results

During the last iteration of the validation process, several results have been achieved. Such results are of worth also for the use cases (mainly the VDI use case). We successfully completed the containerization of two applications selected in the VDI use case (OwnCloud, OpenXchange), demonstrating the better efficiency of container-based SaaS deployment model. In order to evaluate the advantage provided by presented power(energy)-aware resource allocation policy and workload consolidation we tested such algorithms within selected simulators (CloudSimPlus [2] and an in-house developed simulation environment).

1.4 WP6 – LOW POWER SMALL FORM FACTOR DATACENTRE

This work package has continued to maintain the same focuses on integrating a Nallatech designed FPGA-based acceleration board with HPE designed low power, high density server solution. Such integration is also intended to further demonstrate heterogeneity exploitation, and it includes:

1. the designed acceleration card provides both a reconfigurable fabric and a dual-core ARM processor subsystem, which expands the number of processing flavors touched by OPERA;
2. the HPE server chassis (the EL4000 and EL1000 systems) allows to integrate different cartridges, thus enabling the OPERA architecture to use different versions of X86_64 processors

Given the extremely high performance/power ratio of FPGA-based systems along with the availability of large number of I/O resources, WP6 tasks contributed to most of the project's objectives and had a focus on objectives O1 (Heterogeneous architecture), O2 (New generation low power servers), and O9 (Scalable small form factor data center) – see Figure 1. Specifically, **Task 6.1 – FPGA and low power server specification –** aims at studying the specification of the FPGA device (along with the inputs coming from

selected use cases) which is required to be compatible with the HPE servers (Moonshot, EL4000, EL1000), to seamlessly integrate the FPGA board in the server chassis. **Task 6.2 – FPGA design implementation –** focuses on the specific design of the FPGA-based acceleration board (i.e., schematic, layout and firmware), leveraging as much as possible on the features exposed by the HPE server system, and with a strict requirement concerning the PCIe support. A second important objective for this task is the creation of the Board Support Packages (BSPs) which allow the FPGA hardware to be programmed efficiently using the OpenCL tool flow. The main objective and outcome of the **Task 6.3** – **FPGA prototype testing –** is manufacturing the prototype FPGA accelerator card and executing a set of tests to ensure functional design. **Task 6.4** – **FPGA integration on low power server –** objective is to integrate and perform a large set of tests with the manufactured FPGA board and the HPE low power server. To this purpose, the server management board firmware is adapted to enable FPGA board to work on the server. Finally, **Task 6.5** – **FPGA and low power server integration on small form factor data center –** aims at testing that all the communication fabrics are correctly working: (*i*) communication between the HPE server cartridge and the FPGA board; (*ii*) communication through the optical links; and (*iii*) communication between IBM POWER8-based server machine and the FPGA board using CAPI mechanism.

1.4.1 Lesson learned

The lessons learned in WP6 are split in general lessons learned and (if possible) specific for the various tasks. The LL is reported in the following (Table 5):

LESSON LEARNED WP6
General
Offloading of the Approximate Nearest Neighbour (ANN) algorithm as written in the MICMAC source code was found to be problematic for acceleration on the FPGA. Therefore, a brute force version of the code was implemented that allowed a high level of parallelism on the FPGA. It was found that this was also a significantly more accurate implementation.
It was found that the implementation of a Binary Neural Network for the CNN offload work was sufficiently accurate for the YOLOv3 network used. This increased the FPGA performance 4 fold versus a standard floating point implementation.
Task 6.1
Task 6.1 was already completed.
Task 6.2
The new Board Support Package (BSP) significantly improved productivity and resulted in significantly more compact FPGA designs, particularly the ANN offload.

Task 6.3
Final testing of the new hardware design showed no issues with the card.
Task 6.4
During the reported period, the accelerated version of the Approximate Nearest Neighbour (ANN) was completed and tested. Specifically, deliverables D6.8 and D7.6 reported the main results on accelerating such algorithm with FPGA, also comparing the performance with a pure CPU-based implementation (multi-threaded), that means ~10x performance improved versus a single core. The effect increased accuracy of the approach was more difficult to quantify as part of a full MICMAC code.
Task 6.5
We chose to use the ODP feature which has not yet been widely used. This caused problems for our research as we had to ask for a new version of firmware from Mellanox that supported this feature. We had problems later on as well since the performance was not as expected. We have learned to be more careful when using new features that are not widely deployed or tested. While this is sometimes a necessary part of progressing the state of the art, the schedule must reflect the risk involved that something may not work as advertised.

Table 5 - WP6 Lesson Learned

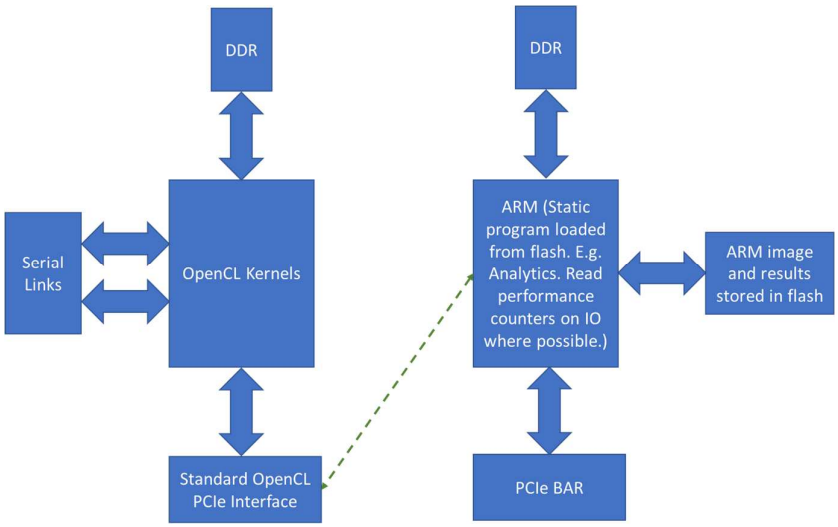

Figure 2 - Final 385A-O BSP organization.

Task 6.1 – FPGA and low power server specification

All the activities related to this task was completed and already reported in other reports of WP6 and summarized in the D2.9.

Task 6.2 – FPGA design implementation

As already reported in D2.10, the use of embedded ARM cores processor on the 385A-O accelerator resulted in very poor performance increase, thus pushing us to use such embedded cores for monitoring purposes and making the programming of the acceleration board easier.

Figure 2 shows the configuration of the new BSP (more details are published in deliverables D6.2 and D6.6). Here, the ARM runs in a standalone mode measuring the power, temperature and any performance counters. During the development of this BSP it was learned that the kernel control signals can be read directly by the ARM from the OpenCL kernel control logic in the FPGA fabric. This provides the ARM with necessary information to generate highly accurate performance metrics of any process running on the

FPGA. Final test in the context of Truck use case demonstrated the ability of measuring performance and power consumption for the ANN algorithm (see also D6.8 and D7.6).

Task 6.3 – FPGA prototype testing

Figure 3 shows the Binary Neural Network (BNN) successfully identify bicycles. It was found that the BNN network was significantly faster on the FPGA and at no discernible cost to accuracy, versus a floating point implementation.

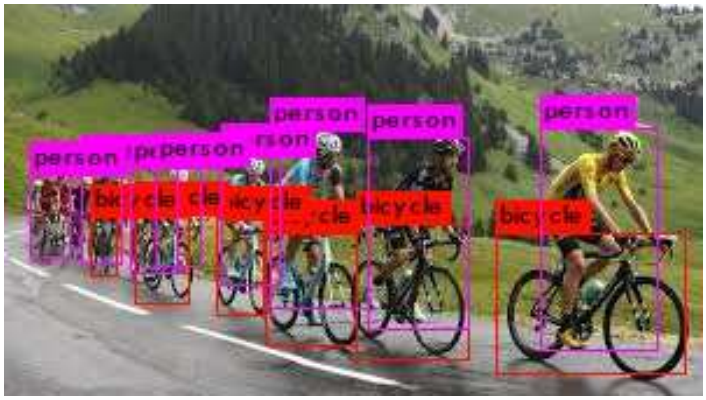

Figure 3 : Binary Neural Network YOLOv3

Investigations into different training techniques of the BNN found that standard "leaky" activations were more accurate for training the network than other forms of activations. See D6.7.

Task 6.4 – FPGA integration on low power server

It was found that the ANN code could be integrated into a single thread of the MICMAC code and provided acceleration in the order of 10x for the equivalent CPU code. This was a significantly more accurate brute force implementation, chosen to allow efficient porting to the FPGA. It was found that the original tree structure of the ANN CPU code did not port well to FPGA technologies.

Parallel Distance Calculations	Peak GOps[1]	DSPs	M20Ks	Registers (k)	Logic Utilization	FMax
12	951	768	610	277	50%	207.81
12 (flattened Loops)	1,001	768	1099	231	40%	218
16 (flattened Loops)	1,433	1024	710	267	45%	234
20 (flattened Loops)	1,478	1280	710	306	51%	193
24 (flattened Loops)	1,792	1536	817	344	56%	195

Table 6 : ANN performance on FPGA

Table 6 lists the different performance metrics for increasing parallelization of the ANN code. See D6.7 for more detail.

Task 6.5 – FPGA and low power server integration on small form factor data center

We were no longer able to integrate the FPGA into POWER8 server platform, since we could not leverage anymore on the vendor support for CAPI protocol implementation. However, to cope with objectives and activities of this task, we integrated a Mellanox NIC supporting the RoCE protocol. The goal was the use of the new hardware for the same application – fast container migration in a heterogeneous data center.

[1] Peak performance is calculated as the number of parallel operations per distance calculation multiplied by the clock frequency

However, the integration of such new board did not come for free, and new challenges we had not considered at the start of the project, such as how to physically connect the Mellanox NIC to the Moonshot system, arose. In addition, we encountered several bugs during development with the Mellanox NIC which caused delays. Working with Mellanox, allowed us to solve all the issues related to discovered bugs, and provide a working solution demonstrating the feasibility of migrating containers.

1.4.2 Achieved results

WP6 activities are generally focused on the design, test and integration of FPGA board within a low power server. In this context, in the reported period several results have been achieved so far. The new BSP for the FPGA acceleration board has been completed and tested. With such new configuration of the FPGA board, we were able to complete the integration of the board on HPE servers and test it. We were also able to complete the implementation of the CNN and ANN algorithm, that are at the basis of the energy efficiency improvements demonstrated by the Traffic monitoring (offloading of the elaboration on the accelerated CNN) and Truck use cases. Regarding the CNN implementation, we were able to pass through a refinement process of the designed solution, which moved from standard implementation of the YOLO CNN architecture, to the one using binary weights (which demonstrated to be more effiecient and better suited for running on the FPGA fabric). Similarly, we completed the design of the ANN algorithm (accelerated version) and tested it, comparing the performance gain against its CPU implementation

In the 3rd cycle, we completed our implementation of remote page faults over RDMA. IBM filed a patent based on this work which uses unique properties of RDMA to reduce the latency of remote page faults. We benchmarked the page fault mechanism using micro-benchmarks and prepared a proof-of-concept implementation to be integrated upstream in the rdma-core subsystem of the Linux kernel. The proof-of-concept was presented at the Linux Plumber's Conference which was well received by the Linux community. We started integration of the RDMA support for remote page faults with CRIU, and also got to the point of a working prototype that could be benchmarked. However, the first implementation was quite poor, and introduced a lot of overhead that needs to be addressed in the next implementation.

2 USE CASES ANALYSIS

In this section we update the status of the three use cases in terms of achieved results so far, encountered issues and difficulties. To this end, here we provide the analysis of the results we achieved during the third iteration of the agile methodology, which is intended to evaluate the gap between expected and achieved results. With respect to the analysis presented in the report D2.10, here we highlight progresses done, specifically regarding the integration of the offloading mechanism in the Traffic Monitoring use case. The gap (if any) between expected results vs. achieved ones constitutes the LL gathered from WP7 and will be the basis for the subsequent analysis performed in Section 3.

2.1 VIRTUAL DESKTOP INFRASTRUCTURE (VDI)

Virtual desktop services (also known as virtual desktop infrastructures – VDI) paradigm moves the computational workload and the power consumption from customer premises to data center. For this reason, it's very strategic for providers to optimize both workload and power(energy) consumption in their data center. The OPERA project aims at proposing a data center oriented solution that: (*i*) support a large base of users per physical machine; (*ii*) reduce the overall power(energy) consumption; and (*iii*) possibly leverage on specialization given by heterogeneous hardware.

2.1.1 Involved WPs

The VDI use case mainly involves technologies and solutions researched and developed in:

- WP4 (methodology and models for measuring energy efficiency);
- WP5 (containerization and migration, workload decomposition and resource allocation in a heterogeneous perspective, virtual memory management).

2.1.2 Analysis of the achieved results

This use case involved both software and hardware aspects of the OPERA solutions we briefly analyse here. A detailed discussion of the iterations of the OPERA solutions validation process is reported in deliverable D7.7, D7.8 and D7.9.

In the last iteration of OPERA design and validation cycle, we focused our attention on the configuration that provided the best performance in terms of Energy Efficiency (EE), that means the containerization technology (LXC/LXD).

Specifically, as reported in D7.9, we considered an additional SaaS application (OpenXchange) and an additional CPU architecture (ARM). Involving these new elements, it was possible to compare the different behaviour in terms of EE according to the application and the CPU architecture, as reported in the following table:

Environment	**EE** (users / kWh per week)
OwnCloud – low power – X86	141,9
OwnCloud – low power – ARM	18
OpenXchange – low power – X86	80,5
OpenXchange – low power – ARM	14,5

Table 7 – EE comparison – X86 vs ARM

We obtained these values, stressing each environment and simulating the workload of virtual concurrent end-users to achieve the 55% of CPU usage, to be as closest as possible to real life loading conditions.

From these results we can understand that is strategic from the provider's point of view, to schedule the application workload properly to improve the EE. For achieving this capability, WP5 developed the following important deliverables:

1. TOSCA descriptor. This is a file describing the composition of software modules being part of the application to run and containing specific TAGs used to pass hints to the orchestrator engine for the subsequent selection of the best hardware to host application components.
2. The energy-aware orchestrator. The development of a software module which runs the power(energy)-aware allocation policies. Such module (namely ECRAE), runs both the greedy strategy used to initially select the nodes where to deploy the application containers, and the consolidation algorithm used to optimize the entire workload allocation on the data center.
3. Container Migration. This is the post-copy migration feature in the CRIU tool between physical systems to realize a dynamic load balancing of applications at run-time.

The TOSCA descriptor consists of three parts:

1. Node types definition. It defines both software component and infrastructural elements (e.g., Storage and Network connection). Through the TAG property, is possible to pass a hint to the allocation engine, in order to later select the best infrastructural resource for that purpose;
2. Relationship types definition. It describes the specific dependencies between different node types, i.e., the connections among nodes and objects;
3. Topology. It contains the information to set up nodes and objects according their connections (i.e., dependencies).

In addition, concerning the TOSCA descriptor, we set up a demonstrator where we used Insomnia client to send the file descriptor to the orchestrator module ECRAE (managing REST API), which in turns communicate with OpenStack for driving the final deployment phase. ECRAE substitutes the TAGs content with the selected flavours (based on the results of the power-aware allocation policy) and translates the descriptor in the internal format (HOT) used by OpenStack deploying module (i.e., HEAT).

For the container migration, we set up a demonstrator where using the CRIU tools we can move a container, from one physical server to another one. More details about these deliverables are reported in D5.8 and D5.9.

In this last cycle, even if we demonstrated the feasibility of both these two features independently, we did not obtain a full integrated solution due to compatibility matrix constraints, mainly the impossibility to manage containers on ARM architecture within an OpenStack-based environment (as reported in D7.9).

2.2 DATA CENTER ON TRUCK

This use case is an example of an instantiation of a small data center, and given the specific purposes for which is intended to:

1. Limiting power consumption;
2. Limiting occupied space;
3. Reducing elaboration time of orthophotos.

For these reasons, it is very important the presence of high-density and high-performance servers and heterogeneous architecture because they allow for improving all the three previous issues. Indeed, the improvement of these three elements could have a great impact for "Protezione Civile", since it is possible to rescue people faster and to operate with the truck for a longer time.

2.2.1 Involved WPs

The data center on a truck use case mainly involves technologies and solutions researched and developed in:

- WP6 (low power servers, FPGA acceleration);
- WP4 (methodology and models for measuring energy efficiency).

2.2.2 Analysis of the achieved results

During the last cycle we focused our effort on two targets:

1. Demonstrate MICMAC porting feasibility on FPGA card;
2. Set up a heterogeneous processing solution usable and useful in real-life operations.

About the first target, as reported in D2.10, accelerating the MICMAC elaboration with the FPGA card required a huge effort due to the need of both a deep analysis of the code and an extensive photogrammetry knowledge. So, in the third iteration, we analysed the MICMAC code and each one of its modules (Tapioca, Tapas and Apericloud). We found out that one of these modules (Tapioca) consumes most of the elaboration cycles (i.e., consequently it consumes large fraction of the execution time), and specifically the ANN command. In the previous design and validation iteration, we identified the ANN-based function as a good candidate for acceleration. After the analysis, WP6 worked on MICMAC code and FPGA porting; the result was one ANN thread accelerated by the FPGA, which performs 24 parallel distance calculations. In these conditions, it was possible to improve performance 10 times and we have only an additional 12% of energy consumption due to the FPGA.

About the second target, as anticipated in D2.10 and reported in D7.9, we set up a second heterogeneous architecture where we substituted FPGA card with Nvidia P4 (GPU card) and the MICMAC with Photoscan Pro. These changes entail a solution usable and useful in real-life operations, as described in D7.9. Specifically, with this second set up we were able to improve the three main original installation issues as described in the following table:

Parameter	SOTA	OPERA
Occupied space	4 U	1 U
Energy consumption	4.2 kWh	0.87 kWh
Processing time	11 h	6 h

Table 8 – OPERA improvements

In the table above, we compare the SOTA service with the second heterogeneous architecture for the three parameters.

2.3 TRAFFIC MONITORING

We investigated two types of traffic monitoring use case

- We investigated the real-time detection of road event or road conditions by ULP embedded video platform. The embedded smartness is expected to optimize the data transfer (and consequently the energy consumption due to communication) and to increase the capability to real time monitor the road network by providing automatic detection of road event.
- We investigated the elaboration of road data off-loading process, where embedded smartness enables to send complex image to the road management centre for treatment by high efficiency server as required.

2.3.1 Involved WPs

The traffic monitoring use case mainly involves technologies and solutions investigated and developed in the following work packages:

- WP3 (ultra-low power processing nodes, reconfigurable antenna, software and algorithms for traffic monitoring application);
- WP4 (methodology and models for measuring energy efficiency);
- WP6 (low power server for remote processing of computational intensive tasks).

2.3.2 Analysis of the achieved results

The experimentation of the real-time detection of road events showed that the OPERA technologies enable to detect efficiency in real conditions both road congestion and wrong way detection.

ULP OPERA solutions enable to improve the energy efficiency at the scale of the embedded device and even more at the scale of the global system (from the field up to the road management centre).

Such an improvement enables to improve very significantly the capacity of monitoring the road network

- The system is able to automatically detect in real time the road events and to real-time transmit an alarm with a high level of reliability: the no detection rate and the false detection rate don't exceed a few %. The detection could be successfully tested in diversified real-time conditions: by day and by night, under sunny or cloudy weather, under rainfall, etc…In each case the smartness which is embedded in the ULP device enable to detect correctly the events.
- The automatic detection of road events and transmission of alarm (with a high level of reliability) enables to monitor in real-time more location, in comparison with the current road monitoring which is based on the viewing of video-stream permanently collected by video surveillance camera: automatic detection and alarm transmission could enable to monitor in real time several ten of site whereas the road operator currently view only 16 images in the same time.
- One major result is the improvement of energy efficiency: as summarized by the following figure 4. Such an energy efficiency improvement should enable to deploy the OPERA devices in relatively compact, light and cheap energy autonomous system. The improvement of energy efficiency is between x30 and x90 for the embedded device. It enables to deploy at large scale the system across the interurban road network, including in isolated location where grids and network are not available, under realistic economical and practical conditions. We can notice that the improvement of the energy efficiency if we consider the global system is even higher (almost three decades) with a larger capacity to real-time monitor the road network as explained above.

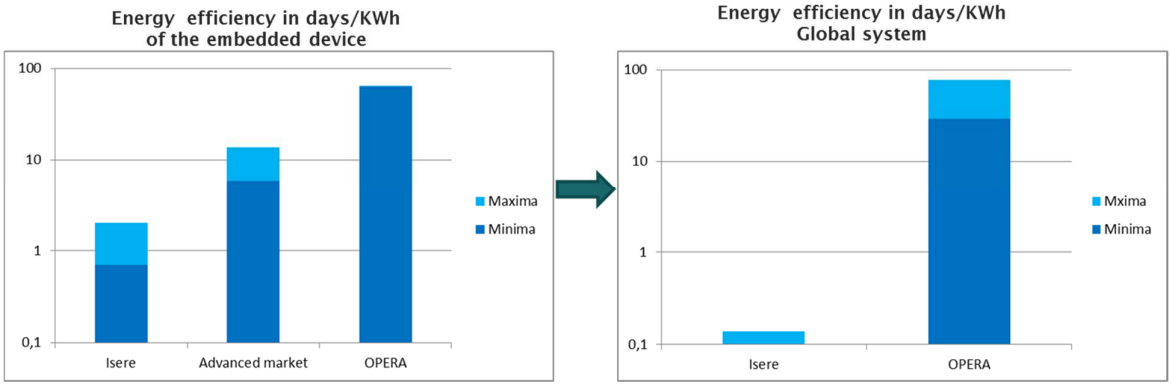

Figure 4 : Energy efficiency improvement for the real-time road event detection

The experimentation of the off-loading process to detect and to count cycles was a successful demonstration of concept: the system implemented successfully a smart transfer of image triggered by the detection of a singularity, and the implemented solution was an ULP solution: the energy consumption of the embedded device is below 0,7 W and the energy consumption for a global system was 97 W for more than 100 cameras. The level of energy efficiency makes realistic the deployment of such solutions across an interurban road network.

3 LESSON LEARNED – LAST ITERATION

Starting from the additional knowledge and experience achieved during the last cycle and reported in the previous sections, in this paragraph we described the results obtained for each technological WP and each UC, highlighting the residual gaps (if any) and their causes.

3.1 TECHNOLOGICAL WORK PACKAGES

At the end of the project, we cannot define new actions to improve the results obtained in the last iteration, so we focus our attention on the residual differences (if any) between target and outcomes.

The scope of theWP3 has been of integrating vision sensing for traffic monitoring applications, and at the same time reaching an adequate level of power efficiency to guarantee enough autonomy. Power efficiency has been reached also by using the offloading feature only sporadically, when the ULP, due to the complexity of the scene, cannot autonomously complete the detection algorithm. For this reason, the software developed in WP3 implemented decision taking solutions, allowing the ULP to decide when the detection could be performed by the ULP itself and when the computation must be offloaded to the remote server.

As an example, the bicycle detection application developed for test site 2 implements, in the Orlando board and in the remote server, two CNNs of the same YOLO family, a PicoYOLO (derived from the TinyYOLO) CNN on Orlando, and a full-featured YOLO on the remote server. When there is a high number of bicycles in the scene (more than 4 or 5 in the same area of the frame), the PicoYOLO can produce only partial results (e.g., miss some of the bicycles) and the full-featured YOLO running on the remote server can be used for obtaining more precise results.

As explained before, WP3 met a series of issues which introduced some delay in the hardware and software development of the platforms to be used in the test sites. Nevertheless, all those issues were solved, with a small impact in the overall WP3 results (less time than planned for testing the wrong way detection and the bicycle detection algorithms). Even if the partners could devote to testing less time than planned, results show to be very good with:

- Very high hardware / software stability of the platform used in test site 1 for traffic congestion detection and wrong way detection (the application can continuously run for months without any problem and with a very high detection quality)
- High performances of the communication link in test site 1 (1-km distance, higher that the .7 kms initially estimated)
- Very high detection quality for test site 2 (bicycle detection), requiring offloading only when a high number of bicycles (more than 4 or 5) are overlapped in the scene (since the bicycles in test site 2 run on a very narrow lane, this situation is very infrequent).

WP4 provides models, methods and algorithms to be able to measure, prove and monitor the energy efficiency of OPERA solutions. One of the main points emerged regards the integration of different technological elements which, if not correctly managed, can prevent to achieve energy efficiency and may make tough to measure the overall efficiency of the proposed solutions. To this end, in the last iteration of the validation cycle, we took care about their integration, in such way the measurement of the energy efficiency was possible. Energy harvesting also represented a key element to make really energy effective the proposed solution. To this end, a more close collaboration of the involved partners allowed to achieve the goal of integrating such components into the working demonstrator. It has been also found important to achieve global OPERA project objectives. On the data center server side, the proposed solution for a more efficient memory allocation mechanism has been found of great importance in the overall context. To this end, although requiring long time, the simulations of memory intensive benchmarks along with proposed hardware software solutions demonstrated essential in order to show the effectiveness of such proposed solutions. Similarly, the simulation of the capabilities of the resource allocation algorithms

(developed within the WP5) demonstrated to be fundamental to achieve the project objectives of reducing the overall power(energy) consumption of a data center. We worked on the full integration of ConnectX-5 board as the interconnection subsystem across heterogeneous nodes, demonstrating its capability of enabling the migration of Linux containers.

Memory management plays a huge role in the optimisation of many applications, especially those that are memory bounded. Despite the availability of larger amount of DRAM on servers, TLB size does not grow at the same pace, and fast simulations of the behaviour of the system running different workload are needed to achieve such optimization (e.g., reducing the energy consumption), thus avoiding usage of slow cycle accurate simulators. To this end, in the last iteration of the validation cycle, we focused on the validation of fast simulation models. In this context we implemented a tool supporting the allocation of memory pages of different sizes, in order to feed the analysed models. Out of completion is the integration of such developed memory allocator with well-known libraries such as 'glibc'. However, the tool developed within WP5 has been of worth in performing energy efficiency measurements.

WP5 is mainly in charge of providing the substrate for efficiently manage heterogeneous resources in the data center. Starting from past lesson learned, we planned actions that were aimed at integrating all the proposed solutions into full working prototypes. As of results of these actions, we successfully demonstrated the capability of the OPERA technology to migrate Linux containers in a fully heterogeneous environment.

Second, we successfully completed the development and test of the infrastructural resource allocation policies in order to demonstrate energy saving over current in use orchestration system. To this end, we simulated for both the static and dynamic allocation algorithms, their behaviour in different conditions, demonstrating their capability of largely reducing the power(energy) consumption of the whole data center.

Work Package 6 deals with the implementation of an accelerator card based on an FPGA device, and its integration into the heterogeneous data centre environment envisaged by OPERA. Using a binary neural network technique, we were able to significantly accelerate the YOLOv3 network on the FPGA accelerator for the traffic monitoring offloading use case. It was discovered that appropriate training techniques of the binary neural network yielded an accuracy more that acceptable for the offload performance. See D6.7. During the development of the ANN code it was found that the structure was not suitable of the FPGA as the CPU version used a dynamic linked list structure. This is not practical on FPGA's as there is no dynamic memory allocation possible. By implementing a brute force approach, (See D6.7), it was possible to accelerate the ANN code, despite the fact that the CPU version of the same code would be 200x slower. By using integer logic, the for distance calculation, it was possible to double the FPGA parallelisation to double the overall performance versus a single precision floating point implementation.

3.2 USE CASES

Use case progress is managed through the activities carried out in WP7. By the analysis of the results reported in specific WP7 deliverables (D7.3, D7.6 and D7.9) and the outcomes of the previous report D2.10 on lessons learned. Here, as done for technical WPs, for each use case we provide a description of the gap (if any) between targets and outcomes.

3.2.1 Virtual Desktop Infrastructure (VDI)

During the last iteration we demonstrated some key elements:

- Each application has a different behaviour, in terms of EE, according to CPU architectures used;
- It is possible to perform cross-ISA migration;
- It is possible to deploy applications following the microservice design pattern and describing them with TOSCA.

Even if we achieved these good outcomes, we did not set up an integrated solution due to the compatibility matrix issues described in D7.9; for such reason it is possible to compare these elements:

- Baseline. At the beginning of the project in D2.1, we declared as baseline the CSI Citrix environment that is a way to implement VDI;
- OPERA solution. The final configuration reported in D7.9 consists of a low power server running Linux containers. Using this configuration, we achieved the best results in terms of EE. During the project, HPE worked on hardware improvements (chassis size and cartridges firmware). IBM defined how to containerize legacy applications, and installed OwnCloud and OpenXchange using Linux containers to demonstrate the capability in a real-life deployment.

Specifically, as reported in the following figure, we have the comparison from the EE point of view:

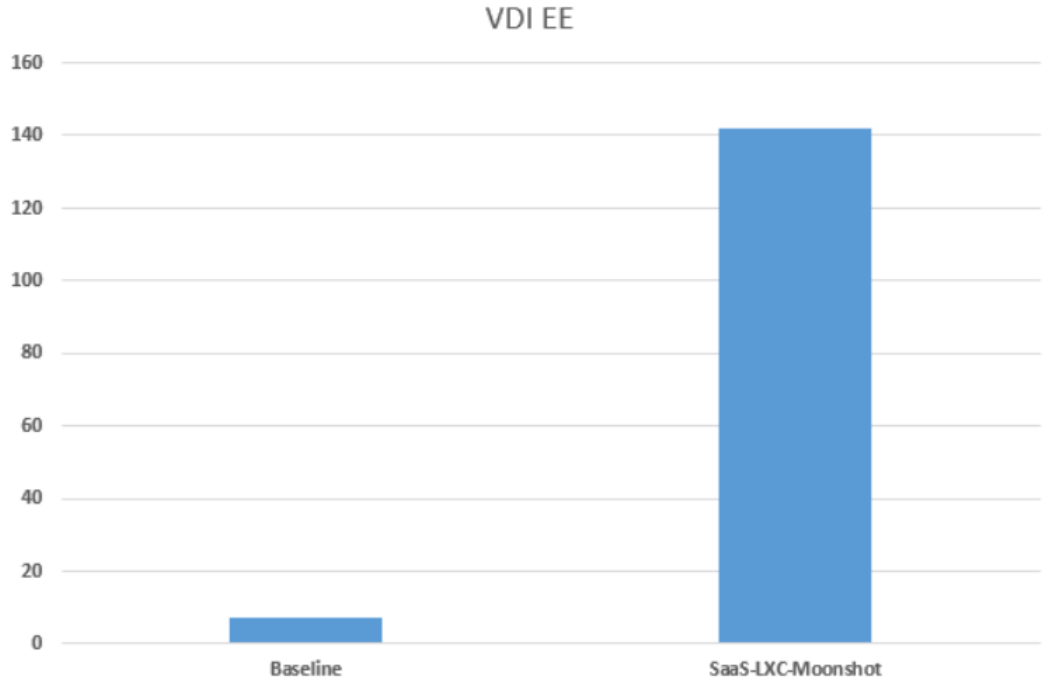

Figure 5 – VDI - EE comparison

We can see that the improvement is 20 times, in fact the baseline EE is 7 and the final configuration EE is 141.9. While this improvement of x20 is quite impressive, a large effort is still required to achieve the original target of x100.

3.2.2 Data center on truck

During the last iteration we demonstrated the feasibility of porting code on FPGA card, but we did not implement a solution usable in real-life operations. However, moving to the second configuration (low power server, PhotoScan PRO, and the GPU accelerator) we found out that this heterogeneous configuration could improve performance and energy efficiency over the SOTA solution. In this configuration the main contribution from OPERA project comes from HPE activities: optimized chassis and cartridges firmware.

For these reasons, we can compare the following two configurations:

- Baseline. In D2.1 we defined as baseline the original installation ((2) HP ProLiant DL380 G5, (1) MacBook Pro 15);
- Last configuration. In D7.6 we described the last configuration (M510 cartridge, Nvidia P4, PhotoScan Pro), that allowed us to achieve the best result. During the project HPE worked on hardware improvements (chassis size and cartridges firmware).

Specifically, this last configuration improved the original installation 5 times as reported in the following figure:

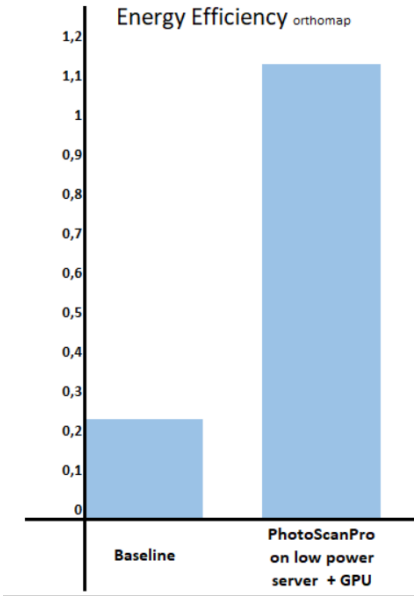

Figure 6 - Truck use case EE comparison

At the end of the project, we can declare that our improvement is x5, which is still far from the original target (improving x100 the EE). Nevertheless, we had two important impacts:

1. We set up on the truck a new service for orthophoto elaboration using the last configuration (M510 cartridge, Nvidia P4, PhotoScan Pro), since with this configuration it is possible to greatly reduce energy consumption (by a factor of 5), elaboration time (by a factor of 2) and occupied space (by a factor of 4) when compared to the baseline configuration, as also reported in D7.6.
2. CSI is setting up orthophoto elaboration service on a second truck that is smaller than the original one.

3.2.3 Traffic monitoring

Concerning the experimentation of the real-time detection, the lesson learned is that the OPERA technology could provide a solution of ULP embedded device to detect road events and to transmit efficiently alarms to the road management centre. The improvement of the energy efficiency makes possible to deploy significantly the devices around the road network. If we combined a more dense deployment of video sensor thanks to the improvement of the energy efficiency and the automations of the road event detection, we can improve very significantly the capacity to monitor the interurban road networks.

The experimentation showed that a commercial product is still expected. The OPERA system still need industrialization step to improve the liability, to improve the hardware packaging and to provide an even more compact and lighter energy autonomous system.

A large scale replacement of the current video surveillance by automatic detection required still to use all the capability of the CNN network to detect large panels of road event or singularities (not only road congestion and wrong way vehicle). It still required a larger demonstration of the liability of these solutions, including in the most critical conditions of the road environment (snow fall for example).

The project shows the practical experimentation required significant work in the field and significant work to improve the liability of the different interfaces between the system.

For the moment ISERE identified these technologies and expects a commercial product. ISERE could be interested to participate to projects about industrialization of the OPERA technologies.

Currently first 4G cameras are currently deployed to transmit permanent video stream, but these cameras are still connected to the electrical grids and only a few cameras can be deployed across a few very strategic location of the road network.

For the road managers the innovation intelligence is essential, about the industrialization of the OPERA technologies, coupled with the possible improvement of the energy autonomous system and of the cellular router. In any case automatic detection of road events or singularities by ULP embedded video devices can be considered as essential for the road managers.

Concerning the off-loading process, the OPERA project enables to prove the concept of the off-loading process based on coupling between ULP embedded video platform (in the field) and high efficiency servers (in the traffic monitoring).

The results are not enough mature to validate the competitiveness of this type of technologies to detect and count cycle as experimented in the project.

But even more important lesson learned is provided by the initially unexpected part of the project: the demonstration of the off-loading process opens a large field of investigation for the road managers. It show that we can collect and local analysis of video data in many site (ULP embedded devices supplied by compact and low cost energy autonomous system) and transmit when necessary transmit images for high level treatment by HPE server in the traffic management center and/or by cloud computing. Considering these results, we become aware that the off-loading process provides:

- The capability to do complex treatment of panel of image collected from many locations.
- The capability to do complex treatment of panel of image collected from many periods.
- The capability to treat complex image.

The road managers have an openness towards new opportunities: many use cases need to be investigated about the road exploitation (detection of road state, prediction of traffic road, etc…), about the road safety (road event detection and analyzing: detection of accident, unadapted behavior etc…) and about the road infrastructure: (monitoring of road surface, civil structure, etc…)

The demonstration of the off-loading process based on ULP solutions both in the field and in the traffic management center open a new field of opportunities that the road managers have to investigate. Combination of ultra-low power technologies, artificial intelligence and cloud computing technologies can modify the road traffic and the road infrastructure management.

4 CONCLUSIONS

The OPERA project dealt with many hardware and software technologies to set up and three use cases involving heterogeneous hardware architectures, aiming to optimize power consumption in real-life operations.

At the end of the project we can claim that the original target of reducing the energy consumption by a factor 100 has been fully achieved in the Traffic Monitoring use case.

- ULP Embedded video platform can detect efficiently road events and road singularities, from energy autonomous devices.
- We proved the concept of heterogeneous architecture combined embedded video platform (in the field) and high efficiency server (in centralized management centre). It opens a large field of investigation to improve the management of the road traffic and of the road infrastructure.

For the other two use cases, although the energy consumption reduction has been more modest, we obtained some good and useful outcomes (as suggestion or starting point for other projects):

- Heterogeneous architecture can reduce power consumption considerably;
- FPGA card could be programmed to intervene in different contexts;
- It is possible to perform efficiently cross-ISA migration;
- We improved the orthophoto processing time in real-life operations.

REFERENCES

[1] Mellanox ConnectX-5, website: http://www.mellanox.com/page/products_dyn?product_family=260&mtag=connectx_5_en_card

[2] Tarun Goyal, Ajit Singh, Aakanksha Agrawal," Cloudsim: simulator for cloud computing infrastructure and modeling", *Procedia Engineering*, Volume 38, pp. 3566-3572, 2012.