Virtual Desktop Use Case 3

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

DELIVERABLE NUMBER	D7.9
DELIVERABLE TITLE	Virtual Desktop Use Case 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.)	01/12/2015
ENDING DATE	30/11/2018
PROJECT WEBSITE	www.operaproject.eu

WORKPACKAGE N. | TITLE	WP7 | Hardware/Software integration and Validation in Real Life workload
WORKPACKAGE LEADER	CSI Piemonte
DELIVERABLE N. | TITLE	D7.9 | Virtual Desktop Use Case 3
RESPONSIBLE AUTHOR	CSI Piemonte
DATE OF DELIVERY (CONTRACTUAL)	30/11/2018 (M36)
DATE OF DELIVERY (SUBMITTED)	15/02/2019 (M39)
VERSION | STATUS	V2.0|Final Version
NATURE	R(Report)/D(Demonstrator)
DISSEMINATION LEVEL	PU(Public)
AUTHORS (PARTNER)	CSI Piemonte

VERSION	MODIFICATION(S)	DATE	AUTHOR(S)
0.1	First draft	19/10/2018	Luca Scanavino (CSI)
0.2	IBM contribution	31/10/2018	Joel Nider (IBM)
0.3	ISMB contribution	05/11/2018	Alberto Scionti (ISMB)
0.4	Certios internal review	15/11/2018	Dirk Harryvan (CERT)
0.5	STM internal review	16/11/2018	Francesco Papariello (STM)
1.0	Final version	16/11/2018	Luca Scanavino (CSI)
1.1	In response to reviewer comments, we reviewed Executive Summary, 1.2.1 and 1.3. In addition, we added also 3.1, 3.2 and 3.3	28/01/2019	Luca Scanavino (CSI)
1.2	HPE contribution	29/01/2019	Gallig Renaud (HPE)
2.0	Final version	31/01/2019	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
CE	Chaos Engineering
EE	Energy Efficiency
KVM	Kernel-based Virtual Machine
ISA	Instruction Set Architecture
LXC	Linux Containers
RDS	Remote Desktop Services
UC	Use Case

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Position of the deliverable in the whole project context

This deliverable describes activities of the VDI Use Case during the third cycle of the Opera project, specifically the period between M29 and M36.

For this reason, the content of the document is closely related to the deliverables D7.7 *VDI Use Case 1* and D7.8 *VDI Use Case 2* in which we described not only the activities and the results we accomplished during the first and second phases of Opera Project (M11 – M28) but also described information and contributions that were important for these results. In this document we do not repeat those points, but we refer to them where necessary, only to promote the understanding of this document.

Therefore, we can consider the considerations about the "Position of the deliverable within OPERA Project" reported in D7.7 also valid for D7.9, because we have the same target, the same partners, the same strategy and the same relationship with the other work packages and other deliverables.

Description of the deliverable

Aiming at guaranteeing coherence and consistency with the other Tasks and Deliverables strictly related to D7.9, we chose the following structure:

- Evolution over the I° and II° Cycles – a summary of the previous phases;
- III° cycle description– Results of the use case during the last phase;
- Outputs and Conclusions – Outcomes due to OPERA project and final remarks.

The main and important inputs for D7.9 are contained in deliverables D7.7 and D7.8 which address these topics:

- The description of OPERA Project targets for each phase
- The configuration of OPERA infrastructure set up during the first and second phases
- The measurements in different VDI configurations and technologies

In these previous documents, we described the experience and the knowledge acquired during the first two phases, but there are also other inputs that can be taken into account: the technological improvements and the measurements indications provided by WP4 and WP5.

In addition, we also report the improvement provided by OPERA not only in terms of energy efficiency but from the knowledge and experience point of view as well and how these can lead to technological change for real services.

In the last chapter (Results & Conclusions), we highlight the improvement in terms of EE comparing the baseline with the best OPERA configuration and the technological items and features that were realized within project effort and we can find in the best configuration.

TABLE OF CONTENTS

1 EVOLUTION OVER I° AND II° CYCLE

In order to better understand the following chapters on the last phase activities and results, we first briefly recapitulate the topics addressed and results achieved in the previous two cycles. Specifically, we report the most significant configurations set up during the first two cycles to highlight and to compare their performance. In addition, we also describe the methodology used to take measurements with the aim to guarantee congruence and uniformity.

1.1 MEASUREMENTS METHODOLOGY

It's very important to define policies and rules for measurements, because during OPERA project we set up different configurations (because there are different ways to realize VDI environments) and for each one we had to compare consistent values.
Specifically, we introduced three different methodologies according the configuration involved to stress test the specific environment:

- for RDS, both open source and commercial product didn't meet performance requirements so we used a lot of tools as described in [1];
- for VDI on KVM-Testbed, we used the proprietary software solution Micro Focus Silk Performer as reported in [1];
- for VDI on KVM-Moonshot and on LXC-Moonshot, we didn't have a Silk Perfomer license for a sufficient number of concurrent end users so we used the discipline called Chaos Engineering (CE) as reported in [2];

The number of concurrent end-users is a very strategic value because the CPU workload and the power consumption are directly related to it. For this reason, to align measurements we adjusted the number of virtual users in each test until a CPU usage of 55%.
In addition, as described in [3], we measured the power consumption and user activity over a week. From these measurements we concluded the fact that we have the following situation for the baseline (as reported in [3]]:

- 36% of the time ("active period") there are active end-users and
- 64% of the time there is an idle condition ("idle period").

Thanks to the number of concurrent end-users and the energy consumption, we obtain the energy efficiency parameter as described in [4]:

$$EE_{vdi} = \frac{1\ week\ full\ operational\ (AU)}{Energy\ used\ during\ 1\ week\ per\ user(kWh)}$$

This value is expressed in user weeks/kWh, the higher this number, the more efficient a configuration performs.

In this way, we have a parameter that considers at the same time a valid period (one week), the number of concurrent end-users and the energy used.

1.2 MEASUREMENTS

1.2.1 Baseline

The baseline configuration is the CSI Citrix environment, as reported in [3], that consists of four virtual machines hosted in a physical server described in the following tables:

Virtual Machines
RAM: it changes dynamically between 4GB and 10 GB
vCPU: 4–CPUs - 4 sockets with 1 core per socket
SO: Windows Server 2008 R2 standard

Table 1 – Baseline description – Virtual machines

Physical Server
Model HP Blade Proliant BL660c Gen8
CPU n°4 x IntelR XR(R) CPU E5-4610 v2 @ 2.30GHz
RAM 256 GB ECC
OS XenServer 6.5

Table 2 – Baseline description – Physical server

In [3], we reported the measurement about the average power consumption during a week for this configuration. We considered a week, because in this period we can appreciate how change the workload due to concurrent endusers. Specifically, as described in 1.1, most of the time (64%) there is an "idle" condition, that means no workload, but servers continue to consume energy (180W). Instead, the rest of the time (36%) we have an "active" condition, when endusers access to services and we consume 240 W. Thanks to that, it's possible to define the average power consumption for baseline:

$$\text{average power consumption}_{\text{baseline}} = 0{,}64\text{week}*180\text{W}+0{,}36\text{week}*240\text{W}=201{,}6\text{W}$$

In [4], WP4 defined the EE considering also the number of concurrent endusers, because the power consumption is strictly related to workload, for this configuration we obtained:

$$EE_{\text{baseline}} = (240 \text{ end-users})/(0{,}2016\text{kW}*7\text{days}*24\text{hours}) = 7 \text{ user weeks / kWh}$$

During OPERA project, we set up different configurations to elaborate the orthophoto, using different hardware and software as reported in [1] e [2]. The EE_{baseline} represents the comparison element to evaluate the results that we obtained during the project for each different configuration. In the last chapter (Results & Conclusions), we'll compared this value (EE_{baseline}) with the OPERA configuration that achieved the best results in terms of EE, highlighting the improvement gained during the project and the technological element developed in OPERA

1.2.2 RDS on Moonshot

The first technology that we considered is the commercial product RDS that we set up with this configuration:

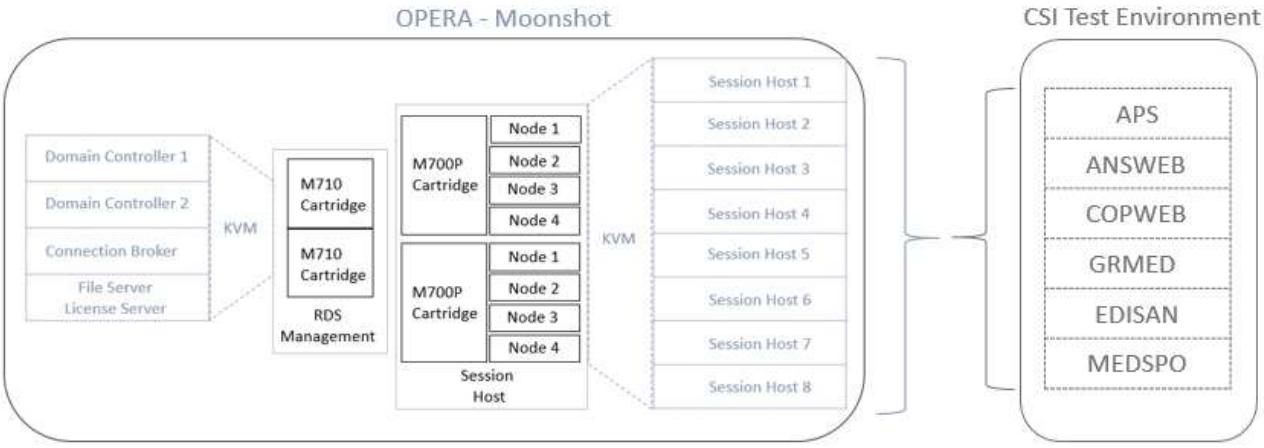

Figure 1 - RDS Environment – Macro Architecture

We installed both the management component (RDS Management) and client nodes (Session hosts) on four Moonshot cartridges as described in [1], instead, the application component is set up in CSI Test environment, because as reported in [1], OPERA takes care only of the end user side.

In this situation, as reported in [1], we obtain this average power consumption and this energy efficiency:

$$\text{Average Power consumption}_{\text{rds-moonshot}} = 75{,}6\ \text{W}$$

$$EE_{\text{rds-moonshot}} = (240\ \text{end-users})/(0{,}0756\text{kW}*7\text{days}*24\text{hours}) = 18{,}9\ \text{user weeks / kWh}$$

1.2.3 SaaS KVM on Moonshot

As alternative of RDS, we introduced SaaS applications installed on Moonshot using KVM technologies, we set up the following configuration:

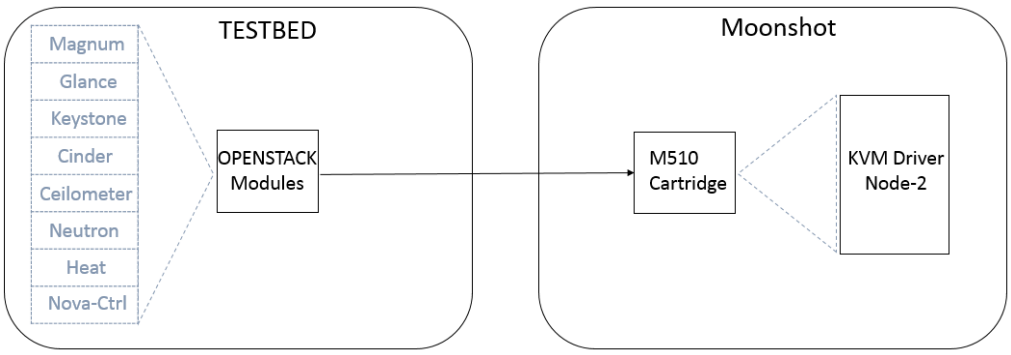

Figure 2 – SaaS KVM on Moonshot – Macro architecture

We can see two servers:

- OpenStack Modules to host all the modules useful for the OPERA project (Magnum, Glance, Keystone, Cinder, Ceilometer, Neutron, Heat, Nova-Ctrl); in this way, it is possible to manage the compute-nodes.
- KVM Driver Node-2 (Moonshot cartridge), where it is possible to host SaaS applications that are split into virtual machines.

More details are available in [1].

In this situation, as reported in [2], we obtain this average power consumption and this energy efficiency:

$$\text{Average Power consumption}_{\text{SaaS-KVM-moonshot}} = 54{,}4\ \text{W}$$

$$EE_{\text{SaaS-KVM-moonshot}} = (1{,}000\ \text{end users})/(0{,}0544\text{kW}*7\text{days}*24\text{hours}) = 109{,}4\ \text{user weeks / kWh}$$

1.2.4 SaaS LXC on Moonshot

In the last configuration, we set up the same SaaS application using the same hardware (one cartridge M510) but involving LXC containers instead of virtual machines. In the following figure, it's possible to see the macro architecture:

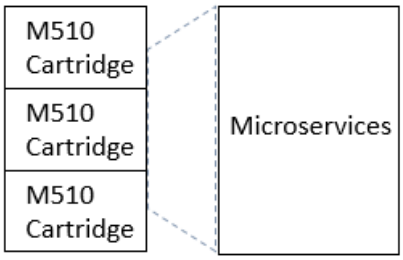

Figure 3 – SaaS LXC on Moonshot – Macro architecture

More details are available in [2].

In this situation, as reported in [2], we obtain this average power consumption and this energy efficiency:

$$\text{Average Power consumption}_{\text{SaaS-LXC-moonshot}} = 58{,}7\ \text{W}$$

$$EE_{\text{SaaS-LXC-moonshot}} = (1{,}400\ \text{end-users})/(0{,}0587\text{kW}*7\text{days}*24\text{hours}) = 141{,}9\ \text{user weeks / kWh}$$

1.3 ENERGY EFFICIENCY TREND

In this section we summarize the outcomes in terms of energy efficiency parameter and we show their values graphically.

In the following table, we report the values described in the previous paragraph:

Configuration	**EE** (users weeks / kWh)
Baseline	7
RDS-Moonshot	18,9
SaaS-KVM-Moonshot	109,4
SaaS-LXC-Moonshot	141,9

Table 3 – VDI Use Case – EE trend

The graphical EE trend is shown in the following figures to highlight the different values achieved for each different configuration set up during the project.

We can see that the technology (hardware and software) can influence a lot the energy efficiency, in fact we have a 20-times improvement when we compare the baseline solution (Citrix environment) and SaaS containerized applications installed on Moonshot cartridge.

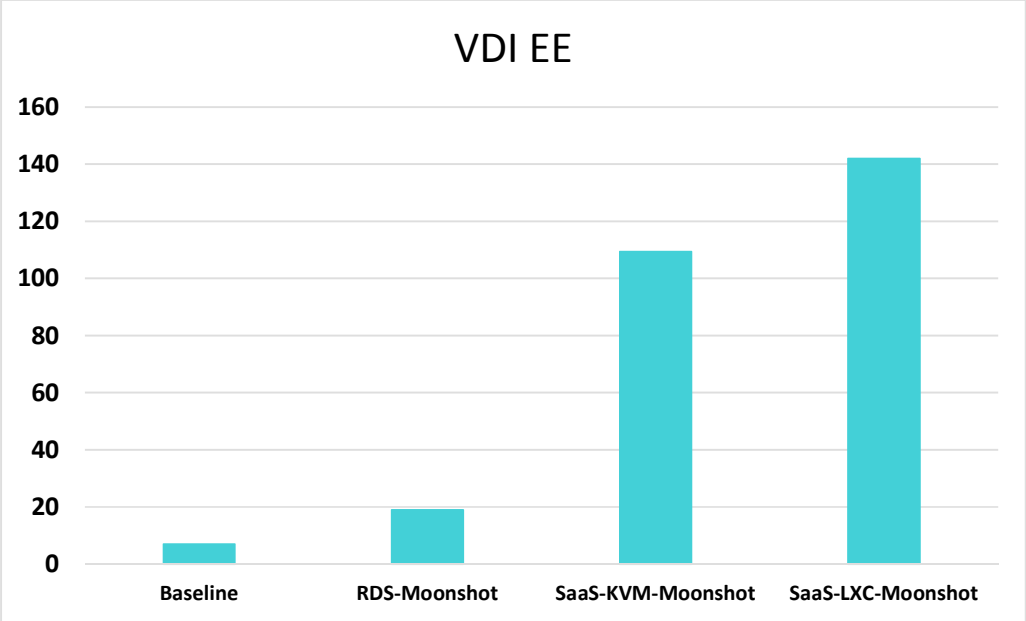

Figure 4 – VDI Use Case – EE trend

Due to these results, we used the last configuration (SaaS application + LXC containers + low power server) during the last iteration described in the following chapter.

2 III° CYCLE DESCRIPTION

In this chapter, we describe the activities completed during the last cycle, that means the measurements about a new SaaS application (OpenXchange) and a new hardware architecture (ARM) to verify the energy efficiency in different condition. In this way, we can compare these results with the measurements about X86 technology. This aspect is very important in relation to WP5 outcomes in terms of microservices migration and Tosca descriptor, so we report the main achievements about that, also because they are the demonstrator topics. More details about that are available in [6] and [7].

2.1 MICROSERVICES MIGRATION

Through the work in WP5, we have developed a method for containerizing legacy application components so they may be deployed in a container cloud environment. The main reason for doing so is to be able to take advantage of management software (such as Kubernetes) which is designed for managing microservice based applications. In this way, the legacy applications can appear as if they are microservices, and can thus be managed in a similar way to born-in-the-cloud applications. Part of this management includes container migration. Although management software such as Kubernetes does not support migration today, we believe that it is a necessary feature that will be supported in the future. In the OPERA project, we rely on migration as a method for dynamic load balancing of applications at run-time. Our load balancing is not aimed at application performance, but rather minimizing power consumption without negative impacts on performance. We developed the post-copy migration feature in the CRIU tool, which is used to migrate containers between physical systems. Enabling CRIU to migrate containers is a necessary first step towards being able to manage containerized legacy workloads in the cloud.

2.2 TOSCA DESCRIPTOR

Microservices design pattern requires the application to be split into independent modules that run on different nodes on top of virtual machines (VMs) or in Linux containers (LCs). Cloud orchestration systems need to know about each of these modules and their requirements in terms of both hardware features (e.g., specific number of CPU cores, amount of memory, etc.) and software (e.g., specific OS version, frameworks, etc.). Cloud orchestrators use specific application descriptor formats to read such inputs and schedule the allocation of VMs/LCs, as well as to set up VMs/LCs with the right software environment. In WP5, we leveraged on a standard description format, namely the TOSCA format, to provide such information to the devised OPERA orchestrator. The TOSCA format has been designed to be agnostic w.r.t. the specific features offered by the orchestration platform; in addition, it offers some degree of freedom to be customized. In this context, we implemented a mechanism that provides information concerning the type of node to use to run the VMs/LCs, thus allowing the orchestration to select the best one in terms of power(energy) saving. The version of the TOSCA descriptor we provided is still complaint with the standard and it is automatically translated in to the specific orchestration format (for the purpose of actual VMs/LCs deployment) by the OPERA orchestrator.

2.3 MEASUREMENTS

During the last cycle we introduced the ARM cartridge (M400) with Ubuntu 16.04 as additional element to the OPERA solution. This is a crucial aspect, because in this way it was possible to investigate new technologies and new configurations. Our goal consisted to integrate the new cartridge as additional Openstack compute node, but despite our effort to achieve this target we weren't able to do that, because of as declared in https://www.ubuntu.com/download/server/arm M400 cartridge supports only Ubuntu 14.04 and not 16.04 as we need and as we installed despite of the support matrix. Due to that, it wasn't possible to replicate the same configuration set up for x86 cartridges (M510 and 710x) that consists of OpenStack-libvirt-LXC, where libvirt is the toolkit to manage platform virtualization. For this reason, it's not possible to introduce the M400 cartridge as OpenStack compute node so we took

measurements using only LXC containers. Specifically, we measured both OwnCloud and OpenXchange containerized on M400 cartridge to evaluate the energy efficiency parameter with this configuration. In addition, we measured also OpenXchange containerized on M510 (X86) to compare the same service using different CPU architecture. We can also do the same comparison with OwnCloud considering the measurements on M510 taken during the second cycle and reported in [2].

In the following figure, we can see the final macro architecture configuration:

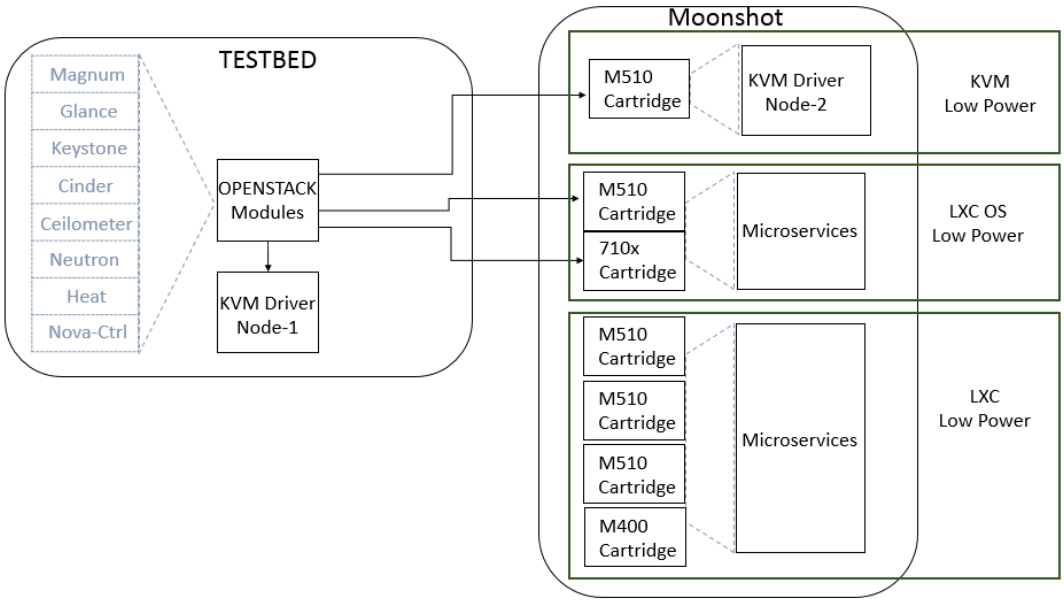

Figure 5 - VDI macro architecture solution

It's important to highlight as reported in [5], that the support matrix constraint combined with Moonshot and CRIU limitation impedes OPERA from realizing a solution that includes at the same time OpenStack, TOSCA and CRIU.

2.3.1 OpenXchange – low power – ARM

In the following figure, we reported the measurements about OpenXchange installed on M400 cartridge:

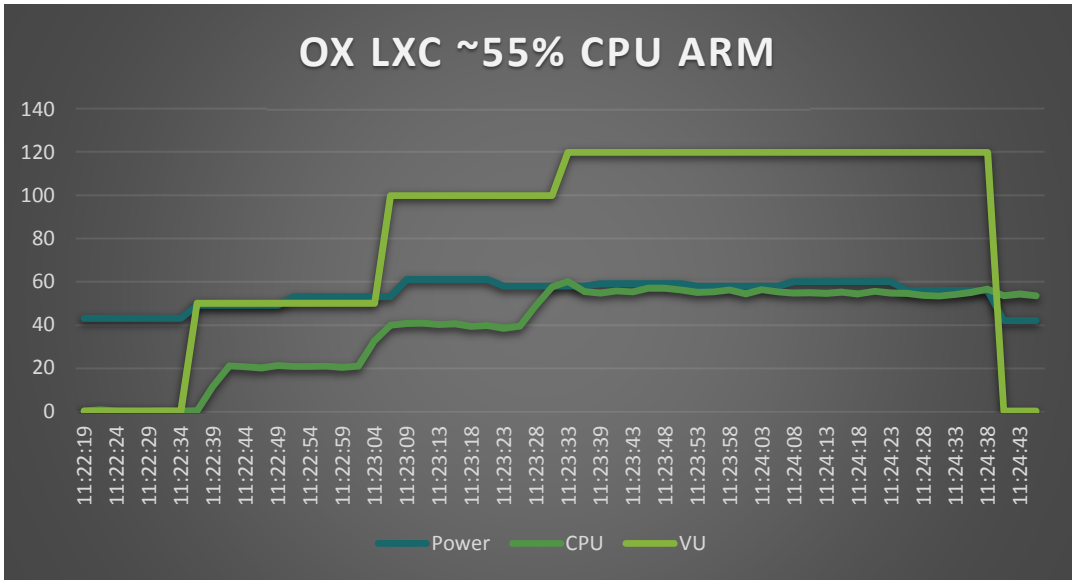

Figure 6 – OpenXchange – low power – ARM

We used the CE discipline to achieve about 55% of CPU usage, in this condition we simulated 120 end-users workload and we consumed 60W. If we add that in the idle condition (no workload) the ARM cartridge consumes 43W it's possible to calculate the energy efficiency parameter:

$EE_{Ox\text{-}ARM}$ = (120 end-users)/((0,043kW*0,64 + 0,06kW*0,36)*7days*24hours) = 14,5 users / kWh per week

In the formula, as described in paragraph 1.1 we considered as active period 36% of the week and 64% as idle.

In Attachments we reported the values measured (Table 5 – OwnCloud – ARM measurements).

2.3.2 OwnCloud – low power – ARM

In the following figure, we reported the measurements about OwnCloude installed on M400 cartridge:

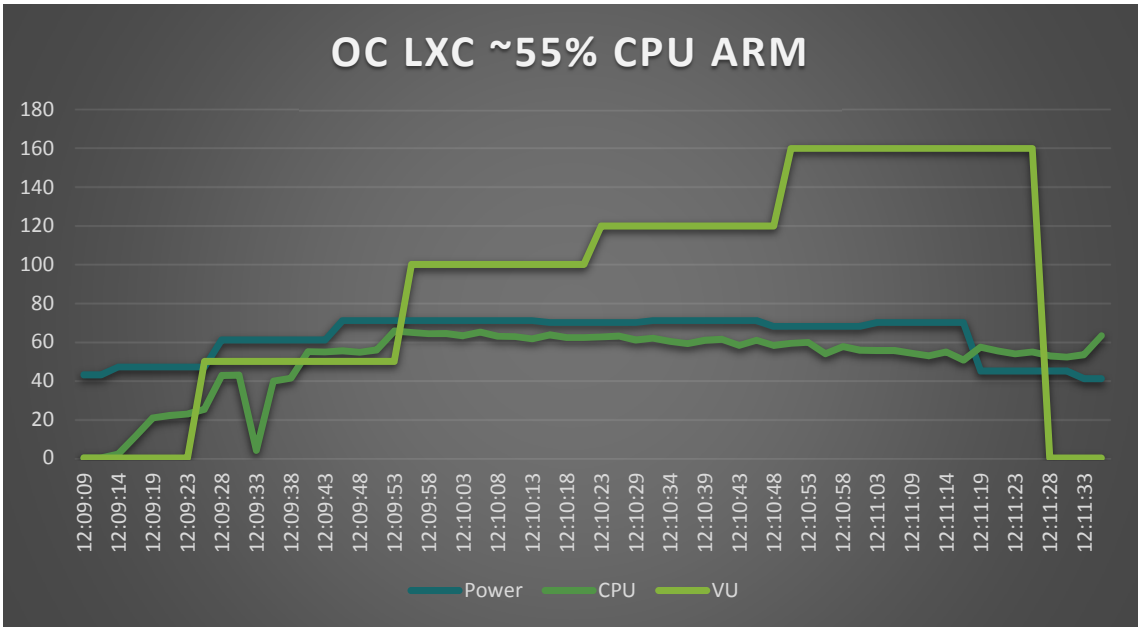

Figure 7 – OwnCloud – low power - ARM

We used the CE discipline to achieve about 55% of CPU usage, in this condition we simulated 160 end-users workload and we consumed 70W. If we add that in the idle condition (no workload) the ARM cartridge consumes 43W it's possible to calculate the energy efficiency parameter:

$EE_{OC\text{-}ARM}$ = (160 end-users)/((0,043kW*0,64 + 0,07kW*0,36)*7days*24hours) = 18 users / kWh per week

In the formula, as described in paragraph 1.1 we considered as active period 36% of the week and 64% as idle.

In Attachments we reported the values measured (Table 6 – OwnCloud – ARM - Measurements).

2.3.3 OpenXchange – low power – X86

In the following figure, we reported the measurements about OpenXchange installed on M510 cartridge:

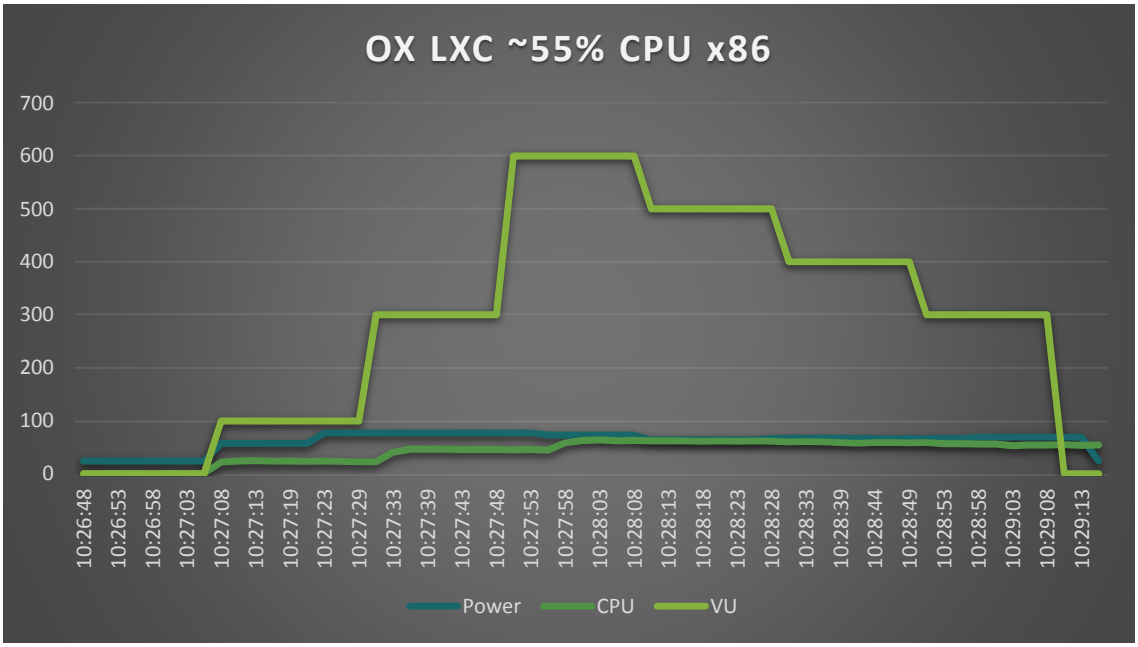

Figure 8 – OpenXchange – low power – X86

We used the CE discipline to achieve about 55% of CPU usage, in this condition we simulated 600 end-users workload and we consumed 77W. If we add that in the idle condition (no workload) the ARM cartridge consumes 43W it's possible to calculate the energy efficiency parameter:

$EE_{OX\text{-}x86}$ = (600 end-users)/((0,026kW*0,64 + 0,077kW*0,36)*7days*24hours) = 80,5 users / kWh per week

In the formula, as described in paragraph 1.1 we considered as active period 36% of the week and 64% as idle.

In Attachments we reported the values measured (Table 7 – OpenXchange – x86 - Measurements).

2.3.4 Measurements considerations

Thanks to the previous measurements, we can compare energy efficiency for different services with different CPU architectures, as reported in the following table:

Environment	**EE** (users / kWh per week)
OwnCloud – low power – X86	141,9[1]
OwnCloud – low power – ARM	18
OpenXchange – low power – X86	80,5
OpenXchange – low power – ARM	14,5

Table 4 – EE comparison – X86 vs ARM

[1] This is the value reported in the table 3 for SaaS-LXC-Moonshot configuration that is the same than OwnCloud – low power – X86

We can highlight these topics analysing the values:

1. the same service has different EE according the CPU architecture;
2. different services have different EE with the same CPU architecture;
3. the ratio between different services changes according the CPU architecture.

These observations are important to show that each service has a different behaviour in terms of energy consumption depending on the hardware, it is very important to know that because this information helps to define the better allocation for microservices (TOSCA descriptor) and to move them (microservices migration) to optimize the energy efficiency according the available hardware.

2.4 DEMONSTRATOR

2.4.1 Microservices migration

We are able to demonstrate post-copy migration of containers using CRIU. This showcases the ability to migrate a container at run-time without suffering from performance loss during the migration. This is possible due to the on-demand nature ("lazy migration") which only performs the minimum of work necessary, when it is necessary. In effect, this delays all work until it is required, which eases the system load and reduces downtime to an absolute minimum. This has been applied successfully to one of the most stringent and demanding fields; real-time gaming. Our demonstrator shows a more business-oriented use case; the Redis in-memory database, which is a commonly used component in web applications.

2.4.2 Tosca descriptor

We are able to demonstrate the capability of the proposed OPERA orchestrator to correctly parse customized TOSCA descriptors and deploy containerized applications on a heterogeneous infrastructure. To this end, we integrated a parser on the orchestrator, that is able to correctly manage added elements of the descriptor file. Specifically, the descriptor files provide, for each VMs/LCs to run, an indication of the type of node required, by using a dedicated field called TAG. It is the responsibility of the orchestrator to find a suitable flavor and accordingly a specific node, starting from the TAG indication. We demonstrated the correctness of the whole deploying chain using two applications: OwnCloud and OpenXChange.

3 RESULTS & CONCLUSIONS

During the project we worked on different technologies (Citrix, RDS, SaaS, containers, TOSCA, CRIU, RDMA, virtual machines, low power servers, OpenStack and so on) and we combined them to analyse different configurations. Thanks to that we demonstrated the better containers performance compared to virtual machines in terms of energy efficiency. For this reason, we focused our attention on containers and WP5 developed a specific software with the abilities to manage containerized application components (fine grained workload management) and to implement dynamic, power-aware scheduling policies.

These outcomes are related to the measurements taken during this last cycle, because we showed that applications, in general, have a different behaviour in terms of power consumption according the hardware involved so it is strategic to define where installing and moving them.

At the end of the project, we didn't realize a solution that includes at the same time all the main topics (as reported in chapter 2.3) because of technological constraints, but we can combine the elements we defined:

- TOSCA descriptor
- Microservices migration
- Measurements of different services in different hardware

to demonstrate the feasibility of what we declared in the project proposal, we show that it's possible to perform the cross-ISA migration. Cross-ISA migration is very important from the energy consumption point of view because, according the hardware technology, services have a different energy efficiency, so it is very strategic to optimize their allocation.

Regardless of not being able to show the full integration of all functionality, we consider the results to be positive, because we are very close to what we declared in [3], in fact before defining the EE parameter our target was:

Average Power consumption 64% *30W + 36%90W= 51,6W

and we obtained as reported in [2] and in paragraph 1.2.4:

Average Power consumption 64% *26W + 36%117W= 58,7W

During the project, WP4 defined in [4] the Energy Efficiency parameter to also consider the number of concurrent endusers, so in [2] we defined as OPERA target:

Energy Efficiency = 145 userweeks / kWh

And we achieved:

Energy Efficiency = 141,8 userweeks / kWh

In the following paragraphs, we compare the best OPERA configuration developed for VDI UC with two aspects:

1. The baseline established at the beginning of the project
2. The OPERA target to reduce 100 times the energy consumption of servers as compared to the state of the art in 2013

Finally, we reported in paragraph 3.3 what we demonstrated during the project and which OPERA features are present in the best OPERA configuration for this UC.

3.1 BASELINE COMPARISON

In this paragraph, we compare the baseline described in section 1.2.1 and the best OPERA configuration. To define which is the best OPERA configuration we can use the *Table 3 – VDI Use Case – EE trend*, selecting the solution that achieve the higher EE, from these values we can declare that the best results using the configuration that involves low power server, SaaS applications and LCX containers.

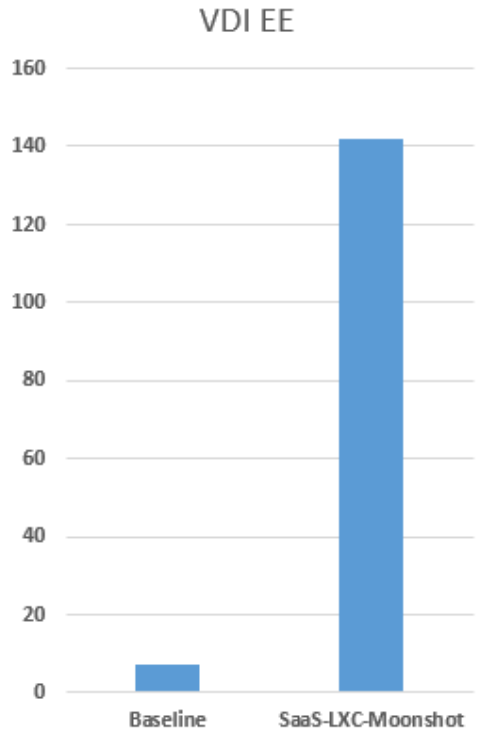

Figure 9 – VDI baseline comparison

Specifically, we improved 20 times the EE, as we can see graphically from the previous figure,

3.2 2013 SOTA COMPARISON

In this paragraph we compare the energy consumed by the best OPERA configuration servers with the state of the art in 2013. The original target was to reduce energy consumption 100 times.

To do that we can compare the average power consumption of baseline configuration with the best OPERA configuration (low power server + SaaS application + LXC containers), these values are reported in chapter 1

$$\text{average power consumption}_{\text{baseline}} = 201{,}6\text{W}$$

$$\text{average power consumption}_{\text{SaaS-LXC-moonshot}} = 58{,}7\ \text{W}$$

From this point of view, the improvements is 3,4 times. But as reported in chapter 1, it's important to consider also the number of concurrent endusers, with the same CPU workload (55%), that means the comparison in terms of EE, described in the previous chapter where we achieved a 20x improvement.

At the end of the project, we can see that we are not close to the original target, the main reason is the difficulties faced to set a full integrated solution with all the functionalities developed in OPERA (cross-ISA migration and TOSCA descriptor).

3.3 OPERA OUTCOMES

During OPERA project, we developed in WP5 two key elements:

- TOSCA descriptor;
- Cross-ISA migration;

demonstrating also their feasibilities. But, as reported before, due to compatibility matrix issues, in the best OPERA configuration we don't have these features but another one developed within this initiative. In particular, in specific stress test configurations (for instance when we add a virtual enduser each second), it's necessary to know the power trend with a granularity of 1 second. So, a firmware has been developed to enable better granularity in the recording of events. By default, HPE only support a record every 5 seconds to avoid premature failures of the NAND flash associated to the management controller (ILO). This has been set as a customized value for Redfish monitoring, with a minimal rate of 1 second.

ATTACHMENTS

OpenXchange – low power – ARM

Date	Power	CPU	Virtual User
11:22:19	43	0	0
11:22:22	43	0,62	0
11:22:24	43	0	0
11:22:27	43	0	0
11:22:29	43	0	0
11:22:32	43	0,13	0
11:22:34	43	0	0
11:22:37	49	0,12	50
11:22:39	49	11,54	50
11:22:42	49	20,93	50
11:22:44	49	20,55	50
11:22:46	49	19,95	50
11:22:49	49	21,13	50
11:22:52	53	20,7	50
11:22:54	53	20,72	50
11:22:57	53	20,85	50
11:22:59	53	20,23	50
11:23:01	53	20,85	50
11:23:04	53	32,67	50
11:23:06	53	39,85	100
11:23:09	61	40,64	100
11:23:11	61	40,8	100
11:23:13	61	40,13	100
11:23:16	61	40,45	100
11:23:18	61	39,22	100
11:23:21	61	39,6	100
11:23:23	58	38,47	100
11:23:25	58	39,39	100
11:23:28	58	48,86	100
11:23:31	58	57,47	100
11:23:33	58	60,12	120
11:23:36	58	55,46	120
11:23:39	59	54,73	120
11:23:41	59	55,77	120
11:23:43	59	55,33	120
11:23:46	59	57,02	120
11:23:48	59	56,93	120
11:23:51	59	56,14	120
11:23:53	58	54,98	120
11:23:55	58	55,24	120
11:23:58	58	56,07	120
11:24:01	58	54,31	120
11:24:03	58	56,23	120
11:24:06	58	55,29	120

11:24:08	60	54,71	120
11:24:11	60	54,87	120
11:24:13	60	54,53	120
11:24:15	60	55,22	120
11:24:18	60	54,38	120
11:24:21	60	55,51	120
11:24:23	60	54,76	120
11:24:25	56	54,59	120
11:24:28	56	53,68	120
11:24:31	56	53,28	120
11:24:33	56	54	120
11:24:36	56	55,04	120
11:24:38	56	56,52	120
11:24:41	42	53,42	0
11:24:43	42	54,27	0
11:24:46	42	53,42	0

Table 5 – OwnCloud – ARM measurements

OwnCloud – low power – ARM

Date	Power	CPU	Virtual User
12:09:09	43	0	0
12:09:11	43	0,06	0
12:09:14	47	2,06	0
12:09:16	47	11,29	0
12:09:19	47	20,7	0
12:09:21	47	22	0
12:09:23	47	22,74	0
12:09:26	47	25,2	50
12:09:28	61	42,69	50
12:09:31	61	42,85	50
12:09:33	61	3,94	50
12:09:35	61	39,66	50
12:09:38	61	41,31	50
12:09:41	61	54,95	50
12:09:43	61	54,84	50
12:09:45	71	55,46	50
12:09:48	71	54,6	50
12:09:51	71	55,87	50
12:09:53	71	65,66	50
12:09:56	71	64,83	100
12:09:58	71	64,24	100
12:10:01	71	64,21	100
12:10:03	71	63,24	100
12:10:06	71	65,12	100
12:10:08	71	62,84	100
12:10:11	71	62,69	100
12:10:13	71	61,66	100

12:10:16	70	63,64	100
12:10:18	70	62,23	100
12:10:21	70	62,29	100
12:10:23	70	62,59	120
12:10:25	70	63,11	120
12:10:29	70	61,07	120
12:10:31	71	61,94	120
12:10:34	71	60,26	120
12:10:36	71	59,22	120
12:10:39	71	60,83	120
12:10:41	71	61,27	120
12:10:43	71	58,21	120
12:10:46	71	60,92	120
12:10:48	68	58,18	120
12:10:51	68	59,37	160
12:10:53	68	59,79	160
12:10:56	68	53,88	160
12:10:58	68	57,64	160
12:11:01	68	55,66	160
12:11:03	70	55,54	160
12:11:06	70	55,61	160
12:11:09	70	54,24	160
12:11:11	70	52,74	160
12:11:14	70	54,77	160
12:11:16	70	50,53	160
12:11:19	45	57,29	160
12:11:21	45	55,3	160
12:11:23	45	53,9	160
12:11:26	45	54,86	160

Table 6 – OwnCloud – ARM - Measurements

OpenXchange – low power – x86

Date	Power	CPU	Virtual User
10:26:48	26	0	0
10:26:51	26	0	0
10:26:53	26	0,06	0
10:26:55	26	0,03	0
10:26:58	26	0	0
10:27:01	26	0,06	0
10:27:03	26	0	0
10:27:06	26	0,03	0
10:27:08	57	21,77	100
10:27:11	57	23,96	100
10:27:13	57	24,96	100
10:27:15	57	23,49	100
10:27:19	57	24,17	100
10:27:21	57	23,12	100

10:27:23	77	24,24	100
10:27:26	77	23,18	100
10:27:29	77	22,73	100
10:27:31	77	22,7	300
10:27:33	77	40,83	300
10:27:36	77	46,13	300
10:27:39	77	46,24	300
10:27:41	77	46,15	300
10:27:43	77	45,78	300
10:27:46	77	45,77	300
10:27:48	77	45,65	300
10:27:51	77	45,25	600
10:27:53	77	45,88	600
10:27:55	73	44,57	600
10:27:58	73	58,23	600
10:28:01	73	62,98	600
10:28:03	73	63,86	600
10:28:05	73	62,44	600
10:28:08	73	62,71	600
10:28:11	64	62,46	500
10:28:13	64	62,43	500
10:28:15	64	61,42	500
10:28:18	64	61,23	500
10:28:21	64	61,54	500
10:28:23	64	61,37	500
10:28:26	64	61,92	500
10:28:28	67	61,06	500
10:28:31	67	59,91	400
10:28:33	67	60,28	400
10:28:36	67	59,85	400
10:28:39	67	58,7	400
10:28:41	67	57,71	400
10:28:44	66	58,61	400
10:28:46	66	58,55	400
10:28:49	66	58,53	400
10:28:51	66	59,1	300
10:28:53	66	56,74	300
10:28:56	66	56,51	300
10:28:58	69	56,17	300
10:29:01	69	55,96	300
10:29:03	69	53,18	300
10:29:05	69	54,01	300
10:29:08	69	54,06	300
10:29:10	69	54,94	0
10:29:13	69	53,61	0
10:29:15	26	54,5	0

Table 7 – OpenXchange – x86 - Measurements

REFERENCES

[1] D7.7 *VDI Use Case I*
[2] D7.8 *VDI Use Case II*
[3] D2.1 *Use Cases and Requirements 1*
[4] D4.1 *Energy efficiency metrics*
[5] D2.2 *Lesson Learned and Track Changes 3*
[6] D5.8 Cloud Software Interface – Final Report
[7] D5.9 Resource Utilization and Allocation – Final Report