Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2016) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

Cloud Futures: From Distributed to Complete Computing, CF2016, 18-20 October 2016, Madrid, Spain

Heterogeneous Secure Multi-level Remote Acceleration Service for Low-Power Integrated Systems and Devices

Lara López[a*], Francisco Javier Nieto[a], Terpsichori-Helen Velivassaki[b], Sokol Kosta[c], Cheol-Ho Hong[d], Raffaele Montella[e], Iakovos Mavroidis[f], Carles Fernández[g]

[a]*Atos Spain SA, Albarracin 25, Madrid, 28037, Spain*
[b] *SingularLogic, Al. Panagouli & Siniosoglou, Athens,Greece*
[c]*Aalborg University Copenhagen, Denmark & Sapienza University of Rome, Italy*
[d]*Queen's University Belfast, University Rd, Belfast, United Kingdom*
[e]*University of Napoli Parthenope, CDN Isola C4, Napoli, 80143, Italy*
[f]*Foundation for Research and Technology – Hellas, Heraklion, Greece*
[g]*Herta Security, Pau Claris 165 4B, Barcelona, 08037, Spain*

Abstract

This position paper presents a novel heterogeneous CPU-GPU multi-level cloud acceleration focusing on applications running on embedded systems found on low-power devices. A runtime system performs energy and performance estimations in order to automatically select local CPU-based and GPU-based tasks that should be seamlessly executed on more powerful remote devices or cloud infrastructures. Moreover, it proposes, for the first time, a secure unified model where almost any device or infrastructure can operate as an accelerated entity and/or as an accelerator serving other less powerful devices in a secure way.

Peer-review under responsibility of the organizing committee of the international conference on Cloud Futures: From Distributed to Complete Computing.

Keywords: HPC; accelerators; devices; offloading; QoS; Cloud computing

* Corresponding author. Tel.: +34 94 242 1007
E-mail address: lara.lopez@atos.net

Peer-review under responsibility of the organizing committee of the international conference on Cloud Futures: From Distributed to Complete Computing.

1. Introduction

Many low-power devices, such as smartphones or tables, as well as several other embedded systems cannot always cope with the increased demand for processing power, memory and storage required by modern applications. As a result, most of these applications are only executed on high-end servers. According to Ericsson Mobility Report[1], there have been 7.3 billion mobile subscriptions in 2015, with an estimation of 5% increase for the next 5 years. Furthermore, the smartphone data traffic is expected to increase by 8.9% in the following 5 years. Users keep demanding more features and services from mobile applications developers. To be able to satisfy users' demands, developers offload the heavy operations of an application to more resourceful machines, such as private or public cloud, giving birth to a new research area known as *Mobile Cloud Computing*[2, 3, 4]. The objective of this paper is to show the approach followed by RAPID[5] project on tackling the above mentioned challenge.

RAPID proposes and designs the full architecture of a heterogeneous offloading framework identifying several components needed for making the system highly automatic and transparent to the developers and final users. Finally, its solution can be applied to many different scenarios, however only three of them will be implemented within the project life: antivirus, hand-tracking and biosurveillance.

Nomenclature	
AC	Acceleration Client
API	Application Programming Interface
AS	Acceleration Server
DFE	Dispatch and Fetch Engine
DS	Directory Server
DSE	Design Space Explorer
RM	Registration Manager
SLA	Service Level Agreement
SLAM	Service Level Agreement Manager
VM	Virtual Machine
VMM	Virtual Machine Manager

2. Architecture

The RAPID system architecture (see **Error! Reference source not found.**) consists of five main components: Acceleration Client (AC), Acceleration Server (AS), Directory Server (DS), Service Level Agreement Manager (SLAM), and Virtual Machine Manager (VMM).

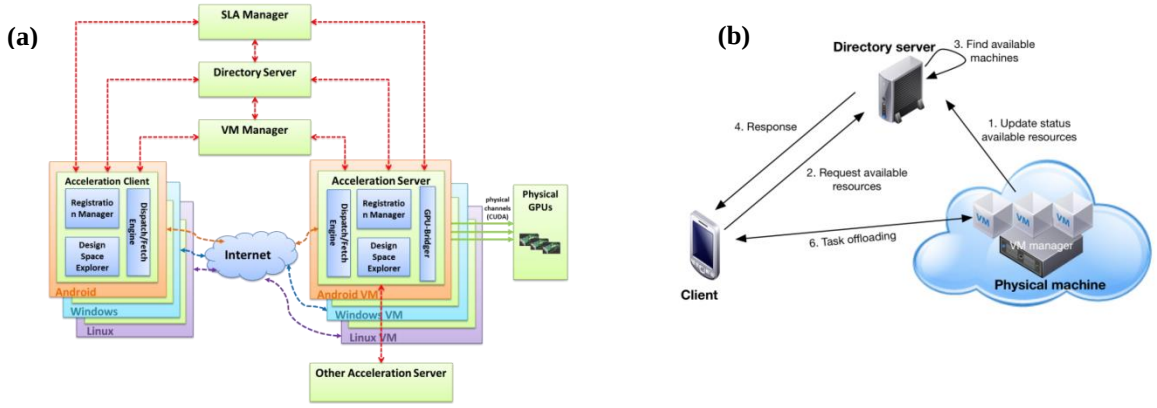

Fig. 1 (a) High-level RAPID logical architecture (b) Interaction between the entities in the RAPID system

On the left side of **Error! Reference source not found.** a) we can see the accelerated devices making use of the RAPID AC to register with the system and to interact with the RAPID AS for task offloading. The AS is a multi-threaded program that is installed in a VM, serving multiple applications of the associated device with remote task execution. To make the discovery, association, and interaction between the AC and the AS transparent to the application developers and to the final users, these steps are handled by the DS, which is responsible for keeping track of all the active actors in the system, and the VMM, which is responsible for controlling the VMs whenever clients come or leave the system. Finally, the SLAM is the component that monitors the quality of service offered to the clients.

Specifically, the role of each component is as follows:

- **Acceleration Client** is the component that handles the task of offloading process on the client side. The AC is composed of a *Registration Manager* (RM) that registers the device with the DS, so that the DS can then allocate a VM for the device. The *Dispatch and Fetch Engine* (DFE) is the only component that the developer is aware of and should use. The DFE handles the execution of the off loadable tasks, running them locally on the device or remotely on the VM, based on the network conditions and each task's execution and memory complexity. The component that decides the execution location of the tasks is the *Design Space Explorer* (DSE), which implements different profilers to monitor the performance of the tasks, the device and the network.
- **Acceleration Server** is the component that runs inside a VM that handles the execution of the offloaded tasks. It also implements a RM, which is used to register the AS with the DS and with the VMM. The AS also implements a DFE, which is the component that interacts with the DFE of the AC whenever a task is offloaded, and effectively runs the task on the VM. Finally, the GPU-Bridger is the component that handles the GPU code offloading, using GPU virtualization techniques through GVirtuS[6]. GVirtuS is a split-driver model-based generic virtualization framework for GPU virtualization, which offers virtualization support for generic libraries such as accelerator libraries, with independence from all involved technologies.
- **Virtual Machine Manager** is the component responsible for managing and monitoring the resources of a physical machine. In RAPID, each physical machine has its own VMM, which provides the information about the resources to the DS, the SLAM and the different AS installed on the VMs running on the physical machine. The component supports not only personal machines but also the usage of well-known Cloud platforms, such as OpenStack[7] and OpenNebula[8].
- **SLA Manager** is a framework developed to be a transversal asset, powerful and flexible enough to be used in several environments and domains. It is based on the WS-Agreement specification and it is meant to cover the following features: Define a language and a protocol for advertising the capabilities of service providers; Create agreements based on the templates; Monitor agreements compliance and violations at runtime; and Take into account a plug-in based implementation, which may provide increased flexibility.
- **Directory Server** centralizes the knowledge of computational resources and active entities in the cloud infrastructure. By this centralization, the DS can inform mobile clients or other components quickly of a list of available physical machines in the Cloud whenever they wish to acquire computational resources in the Cloud. If this physical machine list is given, they can offload, forward or parallelize their tasks to or in available VMs implementing the AS. **Error! Reference source not found.**1 b) shows the biggest actors of the system communicating with each other when performing registration, resource lookup, and task offloading.

3. Challenges and Expectations

The main challenge addressed by RAPID is enabling the mobile cloud computing paradigm by improving the task offloading proposed in previous research works[2]. The project aims not only at providing a quality-based offloading of tasks to remote hosts (by using VMs), but also at increasing the computation capabilities by enabling the possibility to use remote GPUs as a way to accelerate advanced operations required by GPU-capable applications.

Considering that RAPID aims at offering its services to real users, we add new functionalities related to the negotiation and management of SLAs, going beyond the traditional QoS parameters, since the usage of remote

accelerators requires taking into account additional aspects. Furthermore, it is necessary to adequately manage the offloading tasks by improving the allocation mechanisms and the management of available resources.

To this end, RAPID expects to provide the first mechanism for accessing remote GPUs, which will manage resources taking into account applications' requirements and agreed QoS parameters as a way to improve users' experience while, at the same time, it is possible to enable the usage of new applications with high computational requirements in mobile environments.

Moreover, RAPID envisions the implementation of a novel offloading technique between nearby devices, where more powerful and resourceful devices can help the less capable ones in Peer-to-Peer fashion. As it has been shown recently, this collaborative offloading strategy, named Device-to-Device offloading, can be useful in many situations, such as emergency applications, collaborative multimedia streaming, etc.[3, 4].

Security is another challenge to be addressed in RAPID. When data are exchanged between mobile devices and the RAPID infrastructure for offloading, the data can be corrupted or stolen by malicious users. We will overcome this issue by implementing lightweight encryption techniques.

4. Conclusions

This paper presented RAPID, a novel architecture that addresses the challenge of supporting more features and services from mobile applications. RAPID targets offloading of computationally heavy and I/O-intensive tasks from low-power mobile devices to powerful heterogeneous systems in the cloud with multiple CPUs and GPUs. For this purpose, RAPID proposed a new heterogeneous architecture where each component respectively enables task offloading on the client side, handles offloaded tasks in the cloud side with remote CPUs and GPUs, monitors the activity of physical resources, centralizes the information of the resources, and supports the QoS aspects of each user. These implementations will not only realize QoS-based offloading but also create new and more advanced services in mobile applications thanks to powerful remote accelerators. In addition, this novel architecture will provide innovative opportunities to Cloud providers by introducing the concept of *Acceleration as a Service (AAS)*, which opens the door for heterogeneous architecture-based computation in the Cloud.

Acknowledgements

The authors acknowledge the contribution of the whole RAPID consortium. We also acknowledge the support of the European Commission under the Horizon 2020 Program through the RAPID (H2020-ICT-644312) project.

References

1. Ericsson. "Ericsson Mobility Report", June 2016. [Online]. Available at: http://www.ericsson.com/res/docs/2016/ericsson-mobility-report-2016.pdf.
2. Kosta S., Aucinas A., Hui P., Mortier R. and Zhang X., "ThinkAir: Dynamic resource allocation and parallel execution in cloud for mobile code offloading", in *IEEE Infocom*, 2012.
3. Shi C., Lakafosis V., Ammar M.H. and Zegura E.W., "Serendipity: enabling remote computing among intermittently connected mobile devices", in *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*, Hilton Head, South Carolina, USA, 2012.
4. Chatzopoulos D., Ahmadi M., Kosta S. and Hui P., "OPENRP: A Reputation Middleware for Opportunistic Mobile Crowd Computing", *IEEE Communications Magazine, Special Issue on Social and Mobile Solutions in Ad Hoc and Sensor Networking*, 2016.
5. Heterogeneous Secure Multi-level Remote Acceleration Service for Low Power Integrated Systems and Devices, RAPID project, http://www.rapid-project.eu
6. Montella R., Giunta G., Laccetti G., Lapegna M., Palmieri C., Ferraro C. and Pelliccia V., "Virtualizing CUDA enabled GPGPU son ARM clusters", in *Parallel Processing and Applied Mathematics, Springer Verlag*, Berlin Heidelberg, 2016.
7. OpenStack website, https://www.openstack.org/
8. OpenNebula website, http://opennebula.org/