Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 21 (2013) 83 – 90
Procedia Computer Science

The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2013)

Centralized Micro-Clouds: An infrastructure For Service Distribution In Collaborative Smart Devices

Eleana Asimakopoulou[1], Stelios Sotiriadis[1,3]*, Nik Bessis[1], Ciprian Dobre[2], Valentin Cristea[2]

[1]*School of Computing and Mathematics, University of Derby, Kedleston Rd, Derby, DE22 1GB, UK*
[2]*University "Politehnica" of Bucharest, Bucharest, Romania*
[3] *Intelligent Systems Laboratory (InteLLigence), Technical University of Crete, Greece*

Abstract

In the current information-driven society, the massive use and impact of communications and mobile devices challenge the design of communication networks. This highlights the emergency of a new Internet structure namely the Internet of Things that refers to the transformation of physical objects to smart objects and their communication. Based on that the communication of such objects will offer an augmented infrastructure that is formed dynamically and on the fly based on transient links among objects. This is the concept behind cloud computing, to provide a computer-based environment where various services are available to be consumed by everyday users, anywhere and at anytime. Our vision encompasses a dynamic micro-cloud environment that is formed from devices that share computational power. This encompasses inter-linked smart objects and smart mobile devices available from a smart environment that can be formed dynamically. The proposed micro-cloud notion will be of apparent significance to maintain the required quality of service in dynamic scenarios such as those found in emergency and disaster situations. To represent such system we are focused on the development of such architecture into a novel simulation toolkit that allows the replication of Internet of Things scenarios.

Selection and peer-review under responsibility of Elhadi M. Shakshuki

Keywords: Clouds, Micro-clouds, Internet of Things, SimIC, Message Exchanging Optimization.

1. Introduction

Today, the huge utilization of mobile devices has an impact to the development of new networks. In this area the communication of such devices could offer new ways of exchanging data in many forms. Specifically, networks of tomorrow will become data-centric. For example, traffic applications will collect information from traffic, while much of the processing will be done in a powerful data centre such

* Corresponding author. Tel.: +30 6944 271599.
E-mail address: s.sotiriadis@intelligence.tuc.gr

Selection and peer-review under responsibility of Elhadi M. Shakshuki
doi:10.1016/j.procs.2013.09.013

as in a cloud. The cloud computing offers an infrastructure for delivering bespoke service to users via the Internet. There are various types of clouds depending on the service availability and accessibility level. The most common are public, private, virtual public, and hybrid clouds. All these types are directly related with the way that services are provisioned to the end-users. The great advantage of cloud computing is its service elasticity that is the capability of cloud services to increase their capacity dynamically without affecting or need to reconfigure the actual service. As new clients are attracted from clouds innovative features, it has become apparent that clouds require coming together for sustaining quality of service (QoS).

While this transformation from clouds to inter-clouds has shown evidence of high significance, the emergence of Internet of Things (IoT) as a service transforming physical objects to smart objects has further brought both new opportunities and challenges to both cloud and inter-cloud provisions. The IoT visions that embedded intelligence will be fully integrated in everyday life physical objects and in our view, this will involve distributed data been processed, stored and analyzed using distributed processing power and capability available from an available network. In our view, further research is required in the potential embedded intelligence of smart-devices that enable new form of communication and interaction among people and their smart-devices through the utilization of sensors.

In this work, we are built upon on two aspects: a) the computational power that smart-devices enclose and b) the data that smart objects and devices could produce or deploy following a user's request. This will be of significant importance as there are applications where processing cannot happen in a cloud, for example in scenarios where the processing must be done as fast as possible or where the existence or quality of an infrastructure cannot be assumed; such scenarios are those found in emergency and disaster situations.

With this in mind, our vision is that a dynamic micro-cloud encompassing inter-linked smart objects and smart mobile devices available from a smart environment can be formed on the fly. The proposed micro-cloud notion will be of apparent significance to maintain the required QoS in dynamic scenarios such as those found in emergency and disaster situations. This work is organised as follows, section 2 presents the discussion of the research motivation, section 3 the proposition of the micro-clouds model and the simulation toolkit and section 4 the fundamental experimental analysis and its functioning through a use case scenario. Finally, section 5 concludes with a summary of the research findings and the future research directions.

2. Motivation

Cloud is defined as a model in which software is hosted, run and administered in large data centres and provided as a service through Internet. By using a variety of web technologies, both hardware and software Cloud services can be delivered through the Internet in a seamless manner. Several capabilities of Clouds exist, depending on how the Cloud is employed and applied to different areas. The main types of Clouds are: a) Infrastructure as a Service (IaaS), b) Platform as a Service (PaaS), and c) Software as a Service (SaaS). To this extend this study focus is on the micro-clouds and the utilization of IoT smart objects available from a smart environment to form a cloud on the fly.

The proposed micro-cloud contains the scheduling procedures for controlling the service submissions that performed within the cloud in a IaaS setting. Scheduling procedure is defined as the management of the functionalities that directly affect the optimization of issues related with the service submission, distribution, allocation and execution. However, in the case of the proposed system this affects the decision making process for directing the virtualization part of resources. Fundamentally, these issues are related with the generic scheduling functions as happened within a traditional batch system. Within this system, the jobs (services in micro-clouds) are submitted by the IoT users to the resources for execution and form queues.

The utilization of IoT objects as a resource provision mean is particularly important. Authors of [1] suggest that cloud computing comes into focus only when IT finds ways to increase capacity or add capabilities on the fly. This is without investing in new infrastructure, training new personnel, or buying new hardware. They conclude that cloud computing encompasses any subscription-based or pay-per-use service that, in real time over the Internet, extends IT's existing capabilities. Similarly [2] suggest that cloud technologies and models have not yet reached their full potential. There is space for many of the capabilities associated with clouds that are not yet developed and researched to a degree that allows their exploitation to the full degree, respectively meeting all requirements (e.g. heterogeneity, collaboration, information processing as described in [9]) under all potential circumstances of usage. The authors conclude that there are opportunities for provisioning and further development of cloud infrastructures, where in particular telecommunication companies are expected to provide offerings. In [3], authors propose that provisioning and further development of Cloud infrastructures, where in particular telecommunication companies are expected to play a key role. Authors suggest that mobile phones will finally become a true desktop extension and users will be able to communicate and browse services at any time.

The need for mobile phones to be used as the resource means for increasing cloud capabilities has been identified by [4]. Authors present CloneCloud that is a new service that has been developed to move the smart phones to the cloud computing environment. The authors propose that the CloneCloud uses a smart phone's high-speed network connection to communicate with a copy of itself that lives in a cloud-computing environment on remote servers. The clone on the Cloud is much more powerful than the actual smart phone and when the user executes a particular task, it is done either on the phone or on the cloud depending on the resources needed for the task. The CloneCloud uses nearby computers or data centers to speed up smart phone applications. However, authors conclude that clouds need to be evolved to a more a portable fashion by integrating new environment using mobile phones.

This proposed micro-cloud approach moves the state-of-art a step forward by aiming the micro-clouds and the utilization of mobile phones into the cloud by presenting an augmented computational capacity. By moving the smart devices (i.e smart phones) to the cloud computing environment could create a whole new infrastructure that is formed dynamically and on the fly yet similar to large scale distributed system using IoT objects. These particular aspects will be studied during the course of objective 1. A focus on the challenges as well as on the requirements will be done.

In general the IoT has also been described as a paradigm that mainly integrates and enables several technologies and communication solutions. IoT has also been defined as ''a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols.'' Thus, IoT although being a very promising area, yet generated a large discussion based on divergent visions associated with the future paths of IoT paradigm. There have been several proposals for IoT applications especially, as a visionary way to transform current business world as well as to augment the impact on our everyday lives.

In this work we utilize the IoT along with the distributed system in order to form mobile resources in micro-clouds. Resources are linked in order to offer increased elasticity of the micro-cloud users. The IoT concepts will allow the basic unit of the smart object that actually represents an embedded system that ultimately gets connected to Internet thus making IoT as real-time systems. That is to say that embedded intelligence in everyday life objects will allow the distribution of knowledge in the form of processing power and capability within a network. Thus this work proposes to a model architecture to enable the formation of a micro-cloud. The evaluation of the proposed model architecture will be done via a simulation experiment through the use of a generic yet dynamic use case scenario such as those found in emergency and disaster situations. During this part, we assess the usage of 'Simulating the Inter-Cloud' (SimIC) toolkit to enable the simulation experiment. We will specifically focus on assessing that the proposed model architecture enables smart devices to co-work, schedule and offload execution of jobs onto the dynamically formed micro-cloud.

3. The micro-cloud model and simulator

This section presents the discussion of the micro-cloud conceptual model and the description of the micro-cloud simulator. The first part describes the communication part and the dynamic formation of smart devices, while the second part is focusing on the explanation of the key entities of the simulation toolkit.

3.1. The micro-cloud conceptual model

The assumption is that that there is one micro-cloud environment that is formed dynamically from smart-devices. The augmentation of resources is related to the number of interoperable devices each of which is named as sd_a where $sd_a \in SD = \{sd_1, sd_2, \ldots, sd_a\}$ and each sd_a is related to a user u_i where $u_i \in U = \{u_1, u_2, \ldots, u_a\}$. Each micro-cloud is comprised by number of physical smart-resources mc_b where $mc_\beta \in \{mc_1, mc_2, \ldots, mc_b\}$ and $mc_b \subseteq SD$ that constitutes the core computational resources. Further, each sd_a contains a number of physical resources named as pr_γ where $pr_c \in \{pr_1, pr_2, \ldots, pr_c\}$ in such way that the cores, the ram and the storage space of the smart-device integrates the whole device computational capacity. In addition, each physical resource pr_γ generates a number of coordinators namely as mediators $me_e \in \{me_1, me_2, \ldots, me_e\}$ in such way that each device could be in communication with other devices through this component. At last, the assumption is that each of the smart-devices generate a dynamically formed representation of resources (Virtual Machines –VMs) for sandboxing the service requirements. Each $vm_d \in \{vm_1, vm_2, \ldots, vm_d\}$ represents the virtualized part of a smart-device that eventually contains and executes the user service submissions. Figure 1 demonstrates the partnership scenario of micro-clouds where devices could communicate in order to execute a single request.

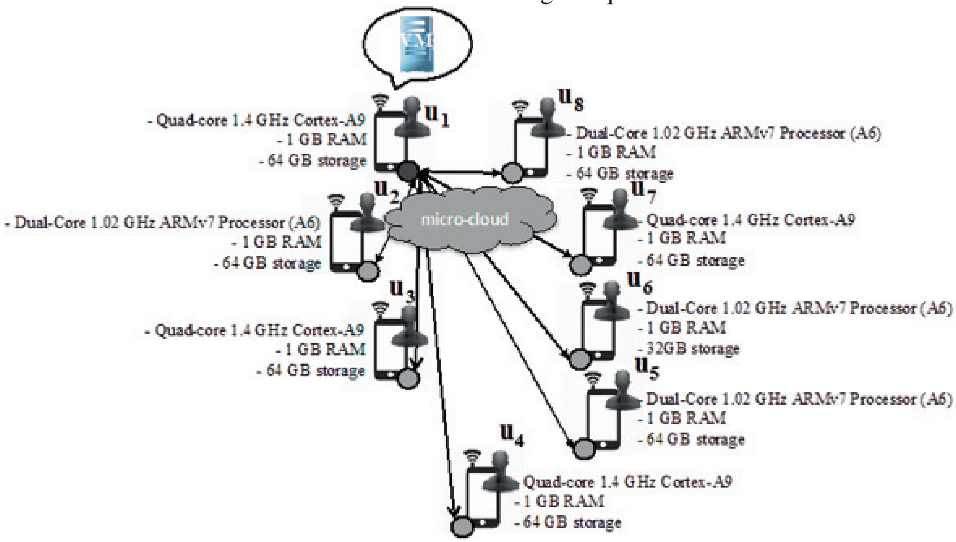


Fig. 1: The micro-cloud formation from smart-devices

In such collaborative setting, a user (part of the micro-cloud community) can submit a service for execution in such way in which each smart device assigns it to the mediator. Then a request for micro-cloud formation is then forwarded to other mediators for managing the exchanging of information among all the smart-devices. Finally, a user (client of the micro-cloud) requests for a service allocation that contains a requirements specification (the job) and passes this request to the mediator. Then the request is distributed to the micro-cloud community where resources form an on the fly VM for job execution. The job is then directly sent from mediator to mediator and sandboxed to the low-level computational resources of the community. This is to say that each job is assigned to a virtual machine that that has been created based on a transient micro-cloud formation.

In detail, figure 1 demonstrates the collaboration scheme of 8 users with their smart devices. At a fist glance, it is apparent that latest mobile devices offer high computational capacities (e.g. Dual-Core 1.02 GHz, 1 GB RAM, 64 GB storage for users 2, 5, 6, 8 and Quad-Core 1.4 GHz ,1 GB RAM, 64 GB storage for users 1, 3, 4, 7), thus making a micro-cloud a powerful cooperative consortium in terms of computational power. It is also shown, that user u_1 request for micro-cloud formation from users u_2 to u_8. Then each mediator receives a request and a resource allocation procedure is taking place. This ensures that the device has the minimum computational power to host a request or a part of the request. The assumption is that the master device of user u_1 splits the task to small parts that could be executed in parallel in the collection of smart resources. At last, each device executes the MEO algorithm to get requests, evaluate resources, rank tasks and respond to master mediator. Finally, user u_1 generates a VM into its local smart device to manage the task executions.

In general, the specification includes that a user requests for resources by establishing connection with a number of smart-devices and by utilizing a job distribution mechanism. This is happening though the micro-cloud mediator that acts as a centralized communication system. Then, the collaboration procedures are operated by the mediator that requests transient formation from the other mobile phones that are in a specific range. Each mediator encompasses a list of smart-devices addresses based on dynamic information extracted from the device (e.g. Bluetooth communication). The assumption is that users are available to receive job requests, allow and send responses for resource availability. Eventually, the micro-cloud will be formed in order to create a VM for executing the request.

The distribution mechanism includes that a request is sent from the master mediator to mediator respectively, and then to the internal resources. After each of which continues with a matchmaking procedure to compare requirements to provisioned resources. Then, each mediator forwarded a message with available resources to the master mediator, only in case of resource availability. After that, the master-mediator collects requests and sends a message to selected resources based on a ranking procedure. As specified previously, our approach follows the MEO model, introduced in [7] and detailed presented in [10]. Further, the resources are allocated and a VM is allocated to the master mediator that distributes the jobs in a parallel-distributed programming mode. The jobs travel to interoperable smart-devices and executed remotely. The extracted results are sending back to the master mediator VM for data integration and analysis.

To conclude, the micro-cloud encompasses a dynamically formed topology of smart-devices wherein various components establish communication with remote ones for exchanging requests for service execution. The fundamental idea is that augmented computational power of smart devices could lead to the transient formation of a virtualized environment. Based on the this study presents the micro-cloud simulator that extends the SimIC toolkit functionality. Specifically, the assumption is that resources are not clusters but smart-devices that could form communication.

3.2. The micro-cloud simulator

The simulation of micro-clouds is based on the classes included into the SimIC simulation toolkit. Principally, this includes a discrete event simulation toolkit based on the process oriented simulation package of SimJava. The SimIC aims of replicating a cloud and an inter-cloud facility wherein multiple clouds collaborate with each other for distributing service requests with regards to the desired simulation setup. Its great advantage is the modular structure that allows developers to easily create or alter default entities and policies.

The micro-cloud simulator extends the default classes of SimIC in order to achieve distribution of jobs among interoperable resources (in our cases the smart devices), This encompasses an automated job distribution among mediators that are located on the top of each cloud in order to communicate with others as in a distributed and interoperable topology. An important issue is the definition of the job specification, thus the simulator uses a Service Level Agreement description file to define jobs. This is readable from all other mediators. The architecture of the simulator involves a variety of intra-cloud and inter-cloud entities as well as supporting classes for service distribution, importing user specifications, exporting performance results and drawing simulation charts. Figure 2 demonstrates the main simulator classes and its relationships.

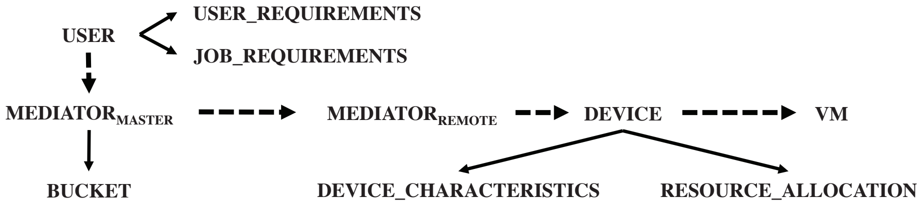


Fig. 2: Relationship of the simulation classes

In particular the architecture of the toolkit includes the extension of the SimIC default classes:

- The **User** class represents the request submitted by the user in the micro-cloud. The simulator allows various requests to be submitted and scheduled after an interval time through the mediator to the micro-cloud community. Further, every submission defines the computational characteristics that are enclosed into an SLA file. The developer is also capable of defining scheduling intervals and latencies. The user requirements are defined as follows:
- The **User_Requirements** class defines the hardware requirements of the user that includes CPU, RAM and storage capacity.
- The **Job_Requirements** class calculates the indicator for the required computational performance measured in million of instructions per second.
- The **Open_Profile** class gets each specific user requirement and passes information to the various micro-cloud simulator entities.
- The **Mediator** class implements the interoperability functionality of the simulator. Specifically, each mediator communicates with others in order to exchange information. The default topology of the toolkit involves a centralized topology. At last each one is linked to a terminal entity (Bucket) that collects requests that have been unable to be executed in order to keep a log of unfinished jobs.
- The **Bucket** class represents the termination entity that collects the unexecuted jobs and logs job profile information for monitoring purposes.
- The **Device** class describes the local resources and the policy for calculating performance in dynamically cases. If a resource is capable of executing the job then it allocates the resource part. The class defines two sub-classes to import device static characteristics (computational resources) and a dynamic policy for resource allocation as follows.
- The **Device_Characteristics** class imports the device static specification that includes CPU, RAM and storage capacity.
- The **Resource_Allocation** class represents the micro-cloud hypervisor that generates VMs. Specifically, this includes a hypervisor queue that places multiple job submissions in a queue. This, as well includes, a scheduling policy for sharing computational power among resources. The default simulation case schedules the queues using first come first serve schedulers.
- The **VM** class encloses the service request to a virtual representation of the smart device. Further, it executes the job and presents the generated results based on the selected metrics. These are the delay, turnaround time, execution time, makespan, throughput and the performance figure.

The aforementioned classes demonstrate the micro-cloud simulator that has been implemented in the SimIC toolkit. A more detailed discussion on the SimIC, its entities' relationships and the meta-brokering dissemination algorithm is presented in [5], [8]. In this study we define a micro-cloud simulator in order to produce the evaluation study and the experimental results of an IoT scenario. The next section illustrates the use-case and the fundamental experimental analysis.

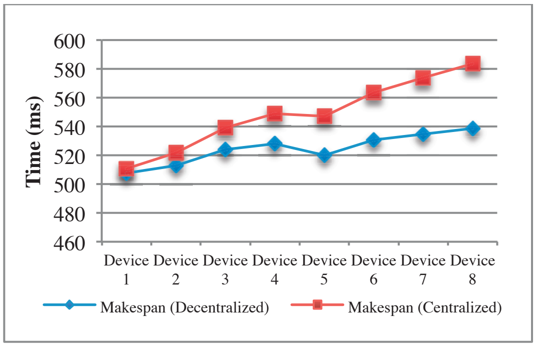
4. A use case for micro-clouds

This section presents a use case of micro-clouds in order to demonstrate the fundamental experimentation. The assumption is that 8 devices with the specification defined in figure 1, collaborate to execute a common aim related to a disaster management. In particular, we simulate 1 job submission from device 1 and we monitor the execution results to become our benchmarks. The disaster management case is related with the analysis and visualization of an emergency situation that is happening in a small area where people mutually agree to allow communication among devices in order to form the micro-cloud. The model utilizes a simple centralized communication pattern wherein user 1 exchanges information directly with all other users. We execute an example that compares the default micro-cloud configuration (centralized) with the Inter-Cloud Meta-Scheduling (ICMS) [6] model that relies upon the distributed scheme. ICMS includes a set of algorithms for job distribution in large scale inter-clouds. The assumption is that information is partial and the services received from the mediators are transient.

This is inspired by the distributed scheme that allows services to be transferred to distant hosts for achieving a performance criterion (e.g. better local resource utilisation, thus leading to global load equilibrium). In view of that, the ICMS utilizes a decentralized architecture for illustrating the service submission, distribution, allocation and execution orientation. This originally defines that each cloud has a local and a meta-scheduler for advanced discussion making. Thus user requests are directly submitted to a meta-scheduler (the user interface) that decides to which local scheduler to relocate it.

The micro-cloud use case is based on ICMS that is composed from a set of algorithms to manage the job life-cycle. By splitting the whole development into sub-modules, we aimed to an efficient management of the dynamic issues of the inter-cloud. The ICMS is formed by a total of five phases namely as a) the service message exchanging, b) the service request, c) the service distribution, d) the service availability, and e) the service allocation. During these phases various components interact with each other, through messages, (e.g. user, local-broker, meta-brokers, datacenters, hosts, VMs) along with several policies for decision-making processes (e.g. resource availability, utilization models, hosts and VMs scheduling and allocation mechanisms etc. implemented in hypervisors).

Based on that, figure 4a shows the comparison of centralized and decentralized cases. It is shown that the decentralized records outperforms the centralized especially for increasing number of communicating devices. Thus, as the time elapses and more devices enter the micro-cloud the decentralization offers better performance. Figure 4b demonstrates the comparison of response times for devices 1-8 between centralized and decentralized cases. Similar to figure 4a, the decentralized setting offers better response times. The high-low lines show that the increasing performance rates as the number of devices increases.



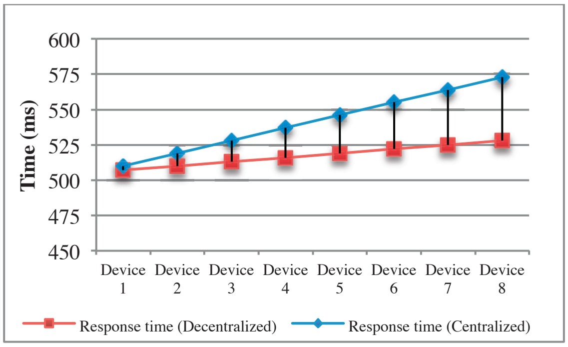


Fig. 3a, 4b: Makespan and resource times comparison for devices 1-8

5. Conclusions and future work

This work presents the dynamic micro-cloud environment that is formed from smart devices that share computational power. The proposed micro-cloud simulator target is to maintain the required quality of service in dynamic scenarios such as the emergency situations for the disaster case. The simulator realizes a simple case where smart-phones communicate with each other in order to execute a common request as described in the experimental analysis. The future steps include the further development of the IoT simulator in order to allow data extraction from sensor devices. In addition the extended experimentation will allow us to define a more detailed discussion of the metrics for dynamic micro-cloud scenarios. An area that could be improved is to allow host scheduling policies to include time-sharing scheduling of multi-cores. Also, although security was treated as out of the scope of the study, exploring the security issues of communication among interoperable micro-clouds will enhance the study. This includes the realization of encryption in entities communication and methodologies for trust management.

References

[1] Knorr, E. and Gruman, G. (2012)' What cloud computing really means', Accessed 28/01/2013, Available at: http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031

[2] Jeffery K., Neidecker-Lutz, B. (2010) 'The future of cloud computing', Accessed 28/01/2013, Available at: http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf

[3] Capasso, V. (2012) 'Cisco: Mobile + Cloud = Future of IT', Accessed 28/01/2013, Available at: http://www.ciozone.com/index.php/Cloud-Computing/Cisco-Mobile-+-Cloud-=-Future-of-IT.html

[4] Chun B-G., Ihm, S., Maniatis, P., Naik, M., and Patti, A. (2011) 'CloneCloud: elastic execution between mobile device and cloud', In Proceedings of the sixth conference on Computer systems (EuroSys '11). ACM, New York, NY, USA, 301-314.

[5] Sotiriadis, S., Bessis, N. and Antonopoulous, N. (2013). SimIC: Designing a new Inter-Cloud Simulation Platform for Integrating Large-scale Resource Management, The 27th IEEE International Conference on Advanced Information Networking and Applications (AINA-2013), March 25-28, Barcelona.

[6] Sotiriadis, S., Bessis, N., Kuonen, P. and Antonopoulous, N. (2013). The Inter-cloud Meta-scheduling (ICMS) Framework, The 27th IEEE International Conference on Advanced Information Networking and Applications (AINA-2013), March 25-28, Barcelona.

[7] Bessis, N., Sotiriadis, S., Pop, F. and Cristea, V. (2012). Optimizing the Energy Efficiency of Message Exchanging for Service Distribution in Interoperable Infrastructures, 4th IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS-2012), September 19-21 2012, Bucharest, Romania.

[8] Sotiriadis, S., Bessis, N. Xhafa, F. and Antonopoulos, N. (2012). From Meta-computing to Interoperable Infrastructures: A Review of Meta-schedulers for HPC, Grid and Cloud, 26th IEEE International Conference on Advanced Information Networking and Applications" (AINA-2012), Fukuoka, Japan, March 26 – 29 2012, ISBN 978-1-7695-4651-3, p.p.: 874-883.

[9] Sotiriadis, S., Bessis, N., Huang, Y., Sant, P. and Maple, C. (2010). Defining Minimum Requirements of Inter-collaborated Nodes by Measuring the Weight of Node Interactions, 4th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS-2010), 15th-18th February, Krakow, ISBN: 978-0-7695-3967-6/10, p.p.: 291-298.

[10] Bessis, N., Sotiriadis, S., Pop, F., Cristea, V. (2013). Using a novel message-exchanging optimization (MEO) model to reduce energy consumption in distributed systems, Simulation Modelling Practice and Theory, Available online 14 March 2013, ISSN 1569-190X, 10.1016/j.simpat.2013.02.003.