Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 21 (2013) 99 – 106
Procedia Computer Science

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

Performance evaluation of interoperable micro-clouds

Stelios Sotiriadis[1,3]*, Eleana Asimakopoulou[1], Nik Bessis[1], Florin Pop[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

The Internet of Things (IoT) is defined as a paradigm transforming physical objects to smart objects that are inter-connected via Internet. Today, IoT objects offer embedded intelligence that can be powerful in case of fully integration of a collective manner towards the satisfaction of user needs. This work is based on the micro-clouds that are a new proposing paradigm to highlight the collective intelligence of IoT objects. Specifically, a micro-cloud could be seen as a pool of cooperated devices and their resources that form transient smart environments. Further to this, we anticipate that the inter-cloud model can expand the micro-cloud capabilities by allowing multiple micro-clouds to communicate in order to achieve a common aim. This will further push the boundaries for studying the interaction and synergetic collaborative nature between micro-cloud systems in terms of their interoperability and performance. However as the size of the system is increased the complexity of performance is additionally increased. This emphasizes the need for decentralization where resources are changing over time without any notice. The vision of this work is that micro-clouds shall be linked together to enable a full network of usable IoT objects and at the same time maintain the required quality of service from an end-user's perspective. Specifically, the aim is to identify the specific criteria which are the most relevant to optimize performance when several micro-clouds collaborate (e.g. load-balancing, throughput, turn-around times, utilization level, etc.) as well as classify their functional requirements. So the focus is on the performance analysis and evaluation of results based on a simulated specific use case scenario.

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

Keywords: Clouds, Micro-clouds, Internet of things, SimIC, Message Exchanging Optimization, Inter-Cloud Meta-Scheduling

1. Introduction

Internet of Things (IoT) is defined as a paradigm transforming physical objects to smart objects that are inter-connected via Internet. The IoT visions that embedded intelligence found in each inter-connected object will be fully integrated in a collective manner towards the satisfaction of user needs. This will lead

* 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.2013.09.015

to a significant distributed data been processed, stored and analyzed using a powerful data centre like a cloud. This will entail a significant distributed processing power been required. More importantly, it is envisaged that end-users and their inter-linked objects can enter and leave from a given smart environment network at anytime. In a similar vein, while end-users may move from one smart environment to another they may have to carry all or leave some of their assets (inter-linked objects) behind.

On another note, research suggests that smart devices (i.e. smart-phones, tablet PCs) will finally become a true desktop extension and users will be able to communicate and browse services anywhere and at any time. There are also evidence that inter-linked smart objects, services and smart mobile devices can form dynamically a mobile-based micro-cloud on the fly. While this is of significance importance, the proposed study is on how to maintain an end-user's inter-linked objects as they move from one micro-cloud to another. While the proposed challenge is analogous to clouds and inter-clouds, one of the significances is defined as in dealing with non-static resources, which at the same time require maintaining the desired Quality of Service (QoS). This involves the interaction of inter-linked services that are all static, all dynamic or a mixture of static and dynamic. Apparently, inter-linked resources could form collaborative clouds.

The concept behind cloud computing is to facilitate an environment for wider distribution of hardware and software based services. Various cloud providers have developed datacentres to offer computational capacity to users that could access services remotely on a pay on demand model. So, cloud computing is defined as a bespoke service setting where resources (hardware and software) that reside to remote locations are utilized by everyday Internet users for a specific time. The on-demand services are included in virtualized environments named as virtual machines (VMs). In this context a cloud defines three main roles namely as the service consumer, the service provider and the service creator. Traditionally, the service creator generates a service that is utilized by the consumer and represents the user hardware and software requirements for leasing cloud capacity. This request is hosted in the premises of the service provider.

A simple cloud service life cycle contains various user requests for services submitted to a cloud service provider. So cloud can be seen as a large-scale environment that combines computing characteristics of various resources (e.g. smart devices). So the job submissions of user tasks share similar features where users could submit requests to a cloud designed from transient formed devices. The overall view covers requests for service (jobs) that submitted by the micro-cloud clients to the service cloud providers. In general, 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. In most of the cases, the cloud architecture shares similar characteristics, thus a micro-cloud could follow the same design. Nowadays, a new model comes to expand cloud capabilities in terms of service elasticity, namely the inter-cloud.

So, the emerged inter-clouds notion forms a pool of collaborated sub-clouds. Thus, the proposed inter-micro-cloud will form a pool of collaborated micro-clouds that are formed dynamically within smart environments. This will further push the boundaries for studying the interaction and synergetic collaborative nature between micro-cloud systems in terms of their interoperability and performance. The complexity of performance is additionally increased when the system is decentralized and resources are changing over time without any notice [9]. Handling sudden variations on huge demands is a critical achievement, thus micro-clouds integration is vital to both realize full elastic services and most importantly, serve individual end-user needs as they move from one micro-cloud to another.

The vision of this work is that micro-clouds shall be linked together to enable a full network of usable IoT objects and at the same time maintain the required QoS from an end-user's perspective. Based on that the paper is organized as follows, section 2 presents a motivation and section 3 a discussion of related works for inter-clouds and inter-collaborated settings. Further, section 4 details the development of a inter-communication model for micro-clouds, section 5 the experimental analysis and the evaluated results based on a simulated specific use case scenario and section 6 the conclusion and future work of this research.

2. Motivation of the study

The inter-clouds concept has been introduced over the recent years as a logical evolution of the Internet. Instead of a file-system oriented Internet, it could be transformed to a computational power oriented setting. Various cloud vendors aimed to an interoperable cloud effort by jointly establishing federations of clouds. However, these vendor-oriented endeavours do not base on future standards and open interfaces. The inter-cloud use cases and functional requirements have been presented in [1]. Authors in [2] discuss that the broker acts as an SLA resource allocator by combining components to achieve the agreed benchmark among users and providers. This is a generic view of brokers that generate questions on how to manage the most effective resource allocation and scheduling.

Specifically, this could produce significant problems when a large number of user requests could cause a system bottleneck. This problem is identical with clouds, in which a new model is required to bridge the gap of resource selection, allocation and scheduling [3]. In contrast to all the above works, we vision an inclusive design of a total decentralized meta-model based on communication among micro-clouds where end-users and their smart-devices are the actual IoT objects. For this purpose, the study proposes the inclusion of a meta-broker on top of each cloud for allowing communication with other meta-brokers during service submission from an IoT object. This means that throughout a request for service execution a meta-broker collaborates directly with other meta-brokers similar to a meta-scheduling system. This will offer significant advantages, as it will support highly interoperability; flexibility and heterogeneity while at the same time an effective service execution setting.

However, in the case of enabling the synergetic collaboration between osculatory micro-cloud, the actual requirements (e.g. computational capacity) are not known in advance and depend on initial conditions and chosen parameters; these are formed during the service submission phase. This is because of the dynamic characteristics of the setting thus a vital requirement is to be orchestrated by a meta-computing component. So the methods for developing a flexible setting should be shelf-adaptive and autonomic based on current decisions, during the run-time, regarding the given initial requirements and conditions. To this extend, our solution aims to achieve dynamic-ness by considering the decentralized meta-brokering paradigm of the Inter-Cloud Meta-scheduling framework [5] by simulating an inter-micro-cloud.

Our approach will imply that a local resource could participate in the on-demand resource selection process at both local and global scale. This is to manage the resource selection, demand allocation and queuing of user tasks (jobs) at a local level by considering the characteristics of the actual system (centralised or decentralised) as well as the temporarily risen requirements of the desired scheduling case. We will deliberate that the meta-computing paradigm, hence meta-scheduling, has proven to be the most appropriate solution, because of its great flexibility when handling the complex requirements of each inter-cooperative system. Consequently, this work contains the critical issues for developing a strategic plan for scheduling job tasks in inter- micro-cloud environments including scheduling, simulation and testing.

3. Related works

The need for highly interoperable settings highlights different kind of demands, either from the perspective of solving complex scientific problems (e.g. in grid computing) or selling and leasing novel e-products to Internet clients (e.g. in cloud computing). This leads to a new need; that traditional scheduling approaches require expanding their functionality based on a range of constraints such as dynamic, heterogeneous, adaptive, and decentralized nature of the environment. Particularly, a mixing of schedulers appears to offer promising solutions because it allows jobs to be shared among different sites (either grids or clouds) [7], thus offering a flexible and scalable setting. Due to that fact, they became an attractive option for high performance computing and information processing. To this extend the term meta-scheduling has proven to be a very important issue in distributed systems [3]. By optimizing computational workloads in the form of an aggregated view the meta-schedulers allows batch jobs to be executed or even migrated to a wider environment by offering the communication bus amongst local schedulers. Specifically, this put forward solutions for bridging the gap between local resource management systems (LRMS-also known as local schedulers).

So, the case of the distributed meta-scheduling theme, which is closely related to the interoperability among resources, originally defines that each resource has a local and a meta-scheduler. Thus jobs are directly submitted to a meta-scheduler and the last one decides to which local scheduler to relocate it. In the simplest of the cases, meta-schedulers query each other at regular intervals so as to collect current load data [3], and to find the site with the lowest load for transferring the job.

This solution is the more advanced and complex, comparing with centralized and hierarchical themes as is more scalable and flexible. Specifically, the meta-scheduler has a partial and instantaneous knowledge of the environment, thus make it a realistic solution for orchestrating large-scale systems behaviour by overcoming issues like bottleneck and single point failures. This work is based on the interoperable micro-cloud settings. Thus, the principal design issue is the decentralization of the system. In addition, the concept of connecting resources together (meta-computing) is at the centre of focus. Thus, scheduling among interoperable micro-clouds can be seen as a meta-computing scheduling mechanism in which a pool of inter-connected resources forming a community. In reality the large scale of the environment exposes a large number of local schedulers that require interacting with each other for optimizing the large number of submitted jobs. This in relation to the fact that resource could change dynamically and could affect the availability of resources makes the system more complex. Thus, having these in mind, the next section describes the model of the interoperable micro-clouds.

4. Modeling interoperable micro-clouds

The interoperable micro-cloud proposes the utilization of a novel meta-scheduling component that conceptually is based upon the meta-computing paradigm. This implies the generation of meta-schedulers by each micro-cloud to handle the service submission requests. For each user the datacenter binds a meta-scheduler that is responsible for controlling requests, SLAs and monitor the whole service allocation and execution. Thus, the meta-scheduler provides an autonomous acting orchestrator that characterizes the initial point of the cloud from the view of the users. The next sections demonstrate the interoperable micro-cloud theoretical model and its algorithmic structure.

4.1. The theoretical model of the interoperable micro-cloud

The interoperable micro-cloud model is based on a meta-brokering solution for job distribution [5]. A user interacts with the broker to request service execution (one or many jobs). The broker acts on behalf of the user and requests specific resources from the cloud system (datacenter). Our architecture is based on the ICMS for cloud service exchanging. Each time a user requests from a cloud meta-broker the request is forwarded to a local-broker that checks for required resources (based on user SLAs). Figure 1 illustrates the ICMS service exchanging between three clouds. The key entities of the ICMS are the users (represented by jobs), the low-level infrastructure (datacenter and hosts) and the resource management components (local and meta-brokers) [6]. The hypervisor include the policy for VM generation in a cloud.

In our case, 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 a component that acts as a centralized communication system. Then, the collaboration procedures are operated by the meta-scheduler that requests transient formation from the other smart devices that are in a specific range. The communication is based on a message model presented in [10].

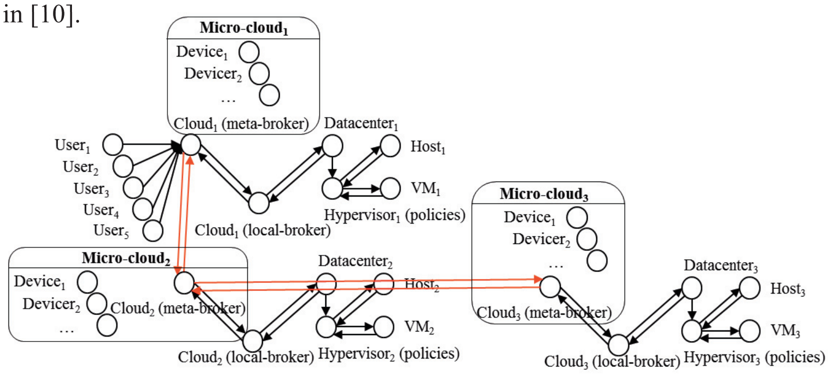


Fig 1. The micro-cloud service initialization

The proposed model assumes that each meta-scheduler has an incomplete knowledge of the actual micro-cloud infrastructure, as the expectation is that during job submission a meta-scheduler communicates with the local resources for exchanging information. This perspective offers a high transparency level for the entire cloud since the users are only mapped to their assigned meta-scheduler, while the last one spontaneously directs the processes to linked meta-schedulers. A micro-cloud contains the normal scheduler (namely as local scheduler) that has a complete knowledge of the cloud infrastructure. In addition, the meta-scheduler is placed on the top of the local scheduler for controlling directly the user requests. For every user (and their services) the local resource generates a new meta-scheduler that directly collaborates with the local and other meta-schedulers of the system. Different from the existing solutions this study realizes inter-cloud by using meta-schedule operations. This moves the complexity of handling service requests from the datacenter level to a component that is harnessed to the actual service submission.

This is an efficient way where, meta-schedulers could identify available resources more easily and reactively check for service execution opportunities. In addition, the meta-scheduler transforms the cloud to an inter-cloud, as it is able to communicate with other resource providers that offer better computational and/or market prospects. Technically, during the service submission, the meta-scheduler communicates with the local scheduler for information exchanging. In addition the meta-scheduler collaborates with other meta-schedulers based on a specification selection of mutual agreed micro-cloud consortium. Next we present the service distribution policy for interoperable micro-clouds based on meta-scheduling characteristics.

4.2. The algorithmic structure of the model

The job distribution algorithm is responsible for the circulation of request for resource availability and delegation from a requesting meta-scheduler in an interoperable micro-cloud. The assumption is that each meta-scheduler instantiates an algorithm in order to communicate with corresponding meta-schedulers belonging to a personal list that is formed based on the accessibility level defined by the micro-cloud community. A use case of an interoperable micro-cloud is presented herein. Specifically, the user request connection to the micro-cloud that assigns the meta-scheduler that starts the procedure.Then the meta-scheduler forms the jobs and exchange messages in order with the other meta-brokers and with the user to collect job information (e.g. the Service Level Agreement-SLA). After, the service distribution algorithm is executed in order to disseminate jobs to the micro-cloud.

Then a VM is generated and into the local resource that executes the request. The assumption is that jobs could be executed in parallel and be hosted by the VM. In detail, each one executes the service distribution algorithm to find remote resources for job executions. The job distribution is based on the Message Exchanging Optimization (MEO) model that is a sophisticated mechanism that optimizes the number of messages. The interactions occur as follows. Each time an interaction occurs, the responding meta-scheduler sends an acknowledgement message to denote the capability of performing the task. It should be mentioned that non-performed decisions are not send. Finally, the requesting meta-scheduler performs a decision making process to evaluate the acknowledgement messages and sends a service allocation request.

Specifically, we assume a meta-schedulerα that requests for resource availability from a group of meta-schedulers belonging to a personalized list. Each of the meta-schedulers is responsible for getting availability figures from local resources (smart devices computational power). This happened by evaluating the shared SLA and job specification as posed by the user. The responding meta-schedulers check internal resource availability and send acknowledgement message (with regards to level of availability) back to the initial meta-scheduler. Finally, the last one decides the best available resource based on a criterion and requests the execution of the job in the selected smart device. At last, a deployment acknowledgement message is send back to the requesting meta-scheduler denoting that the resource is ready to accept utilization by the user. The last step is the utilization of the VM in the resources of the requesting meta-broker. Finally, the results of the jobs that have been scattered to the interoperable micro-cloud are returning back. The algorithm micro-cloud service distribution illustrates the service distribution pseudo-codes.

It is demonstrates that a job profile it is formed from the actual service submission (serviceα) and an SLA specification. For this study the SLA includes a list of physical characteristics such as CPU, CPU cores, memory, storage size and bandwidth. This forms the request of a user of smart-device for interoperable micro-cloud formation. Then, the request is forwarded to the meta-schedulers that are linked with the requester. It should be mentioned that each meta-brokers could be connected with others in different layers due to decentralization. Thus for all meta-schedulers from a meta-scheduling list a request for service availability is sent (in the form of acknowledgement). Then, the SLA comparison defines whether resources are available and a temporary list of non-violate meta-scheduling addresses is formed. Finally, the resource selection criterion is executed that defines the metric (e.g. execution time) for final resource selection. The service allocation mechanism (introduced in [5]) is then executed.

Algorithm: Micro-cloud service distribution

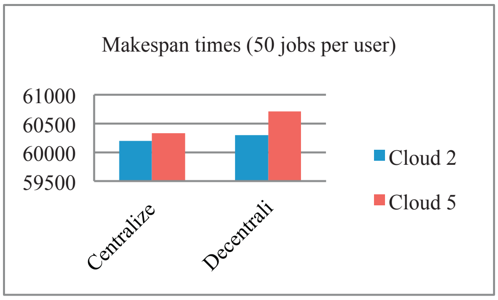
```
jobProfile_α ←(service_α, SLA_α)
for all meta-scheduler_v from meta-schedulerList  do
     meta-scheduler_v run ServiceAvailability for jobProfile_α
     request ServiceAvailability ack from meta-scheduler_v
     if SLA_v = SLA_α then
          SLAViolation = FALSE
     enf if
     temporary-meta-schedulerList[counter] ← meta-scheduler_v
     count++
end for
for all meta-scheduler_v ∈ temporary-meta-schedulerList[] do
     run ResourceSelection on criterion
     selected-meta-scheduler ← temporary-meta-schedulerList[ResourceSelection.count]
     end for
send request selected-meta-Scheduler → run ServiceAllocation
send response selected-meta-Scheduler → meta-scheduler_α.ack
selected-meta-scheduler.wait
```

The next section presents the experimental analysis and the preliminary results of the interoperable micro-cloud.

5. Experimental analysis of interoperable micro-clouds

The experimentation is based on a micro-cloud simulator that extends the default classes of SimIC [4] 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. The experiment demonstrates comparison for high workload submissions (50) per user in an interoperable micro-cloud (centralized versus the decentralized interoperable micro clouds).

In order for the results to be comparable we associate clouds with exactly the same utilization levels (e.g. clouds 3 and 4 with utilization of 20% and clouds 2 and 5 with utilization 6%). Jobs that cannot be executed due to non-resource availability or SLA mismatching are dropped, as the dynamic workload is inactive. Figure 2 shows that the makespan times for 50 jobs per user are significantly optimized results for decentralized example and clouds 3 and 4 (same utilization levels). The average makespan time for centralized case (clouds 3 and 4) is 639706 ms while the same metric value for decentralized (clouds 3 and 4) is 638806 ms (900 ms difference). Figure 3 shows the makespan for clouds with low utilization of 6% (clouds 2, 5).



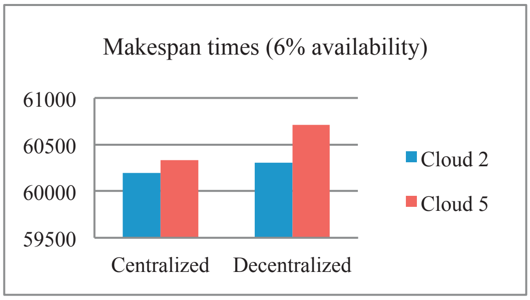


Fig. 2, 3: Makespan times for 50 jobs per user for clouds with same utilization (clouds 3, 4), makespan times for 50 jobs per user for clouds with same utilization of 6%

The next scenario is based on the Inter-Cloud Meta-Scheduling (ICMS) model [5] that provides a set of algorithms for job distribution in inter-clouds. In addition, we utilize the Message Exchanging Optimization (MEO) model [10] to allow a more sophisticated message exchanging solution.

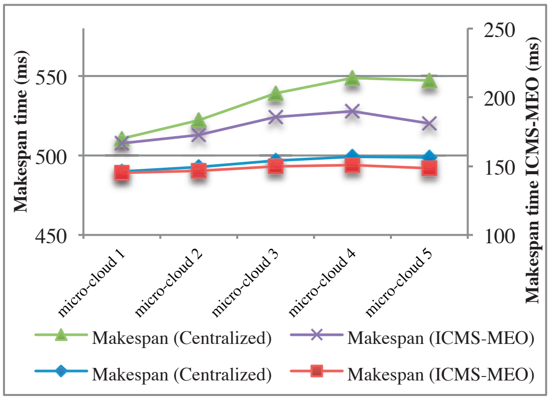


Fig 4. The makepan times for interoperable micro-clouds

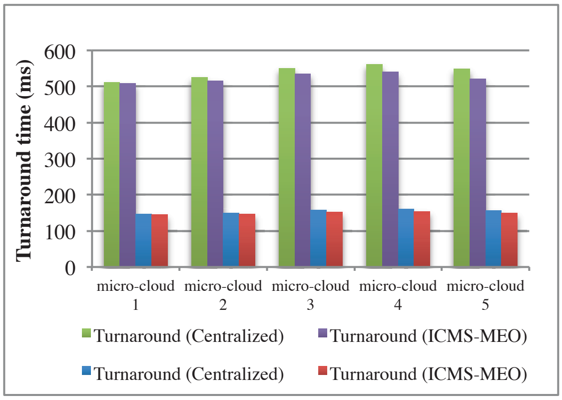


Fig 5. The turnaround times for interoperable micro-clouds

The scenario describes an interoperable micro-cloud of 5 clouds that exchange messages for job distribution. The experiment specification includes 5 devices that share CPU capacity of 6 GHz, 5 GB RAM and 160 GB storage disc. Figure 4 shows the makespan times for a centralized micro-cloud and the comparison to a solution integrated using the ICMS-MEO model. The makespan defines the total length of the schedule and the turnaround time is the total time from submission to execution. It is shown that the makespan times have been well optimized for the ICMS-MEO case. Similarly, figure 5 demonstrates that the ICMS-MEO model outperforms the figures of the centralized case.

6. Conclusions and future work

This work presents the interoperable micro-clouds that are linked together to enable a full network of usable IoT objects and at the same time maintain the required quality of service from an end-user's perspective. The fundamental experimental analysis, demonstrates the usage of existing models for inter-clouds could offer additional capabilities towards to interoperable micro-clouds. The future research steps include the further experimentation and realization of the model into the module of the SimIC simulator. In addition, we will detail the low-level architecture of a specialized use-case (e.g. configuration of smart resource) in order to define specific metrics and extract further quantified results. A future step is to include a more advanced solution to further improve results with regards to resource allocation, and service execution processes.

References

[1] Global Inter-Cloud Technology Forum, (2010) Use Cases and Functional Requirements for Inter-Cloud Computing, GICTF White Paper , August 9, 2010, accessed January 15 2013: http://www.gictf.jp/doc/GICTF_Whitepaper_20100809.pdf

[2] Kertesz, A., Rodero, I., and Guim, F. (2008) Meta-Brokering requirements and research directions in state-of-the-art Grid Resource Management. Proceedings of the CoreGRID Integration Workshop 2008, April 2008, pages 371-382.

[3] Bessis, N., Sotiriadis, S., Xhafa, F., Pop, F. and Cristea, V. (2012). Meta-scheduling Issues in Interoperable HPCs, Grids and Clouds, International Journal of Web and Grid Services, InderScience Vol. 8, Issue 2, Inderscience, ISSN: 1741-1106, pp.: 153-172.

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

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

[6] 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.

[7] Bessis, N., Sotiriadis, Cristea, V. and Pop, F. (2011). Modelling Requirements for Enabling Meta-Scheduling in Inter-Clouds and Inter-Enterprises, Third IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS-2011), Fukuoka, Japan, November 30 – December 2, 2011, ISBN: 978-0-7695-4579-0, p.p.: 149-156.

[8] 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.

[9] Sotiriadis, S., Bessis, N., Huang, Y., Sant, P.a Maple, C. (2010). Towards to Decentralized Grid Agent Models for Continuous Resource Discovery of Interoperable Grid Virtual Organizations, International Workshop on Distributed Information and Applied Collaborative Technologies (DIACT-2010), in conjunction with the 3rd International Conference on the Applications of Digital Information and Web Technologies (ICADIWT-2010), 12th -14th July 2010, Istanbul, p.p.: 170-175.