Meta-scheduling issues in interoperable HPCs, Grids and Clouds

Nik Bessis*, Stelios Sotiriadis

School of Computing & Maths,
University of Derby, Derby, United Kingdom
E-mail: n.bessis@derby.ac.uk
E-mail: s.sotiriadis@ derby.ac.uk
*Corresponding author

Fatos Xhafa

Departament de Llenguatges i Sistemes Informàtics,
Universitat Politècnica de Catalunya, Barcelona, Spain
E-mail: fatos@lsi.upc.edu

Florin Pop, Valentin Cristea

University "Politehnica" of Bucharest,
Bucarest, Romania
E-mail: florin.pop@cs.pub.ro
E-mail: valentin.cristea@ cs.pub.ro

Abstract: Over the last years, interoperability among resources has been emerged as one of the most challenging research topics. However, the commonality of the complexity of the architectures (e.g. heterogeneity) and the targets that each computational paradigm - including HPC, grids and clouds - aims to achieve (e.g. flexibility) remain the same. This is to efficiently orchestrate resources in a distributed computing fashion by bridging the gap among local and remote participants. Initially, this is closely related with the scheduling concept which is one of the most important issues for designing a cooperative resource management system, especially in large scale settings such as in grids and clouds. Within this context, meta-scheduling offers additional functionalities in the area of interoperable resource management, this is because of its great agility to handle sudden variations and dynamic situations in user demands. Accordingly, the case of inter-infrastructures, including InterCloud, entitle that the decentralised meta-scheduling scheme overcome issues like consolidated administration management, bottleneck and local information exposition. In this work, we detail the fundamental issues for developing an effective interoperable meta-scheduler for e-infrastructures in general and InterCloud in particular. Finally, we describe a simulation and experimental configuration based on real grid workload traces to demonstrate the interoperable setting as well as provide experimental results as part of a strategic plan for integrating future meta-schedulers.

Keywords: cloud computing; grid computing, meta-scheduling; InterCloud, scheduling in interoperable infrastructures, dynamic meta-scheduling

Biographical notes: Nik Bessis is currently a Head of Distributed and Intelligent Systems (DISYS) research group, a Professor and a Chair of Computer Science in the School of Computing and Mathematics at University of Derby, UK. He is also an academic member in the Department of Computer Science and Technology at University of Bedfordshire (UK). His research interests have particular focus on the study and use of next generation including grid technologies and inter-cloud computing methods for the benefit of various virtual organizational settings. Nik has successfully involved in (£2m) and led (£0.6m) of funded research and commercial projects in these areas. Prof. Bessis has published over 120 papers, won 2 best paper awards and is the editor of several books and the Editor-in-Chief of the IJDST. In addition, Prof. Bessis is a regular reviewer and has served several times as a keynote speaker, conferences/workshops/track chair, associate editor, session chair and scientific program committee member.

Stelios Sotiriadis is a Graduate Teaching Assistant, a PhD research student and a research associate member of the Distributed and Intelligent Systems (DISYS) Research Group of the University of Derby.

Fatos Xhafa holds a PhD in Computer Science from the Department of Languages and Informatics Systems (LSI) of the Technical University of Catalonia (UPC), Barcelona, Spain. He was a Visiting Professor at the Department of Computer Science and Information Systems, Birkbeck, University of London, UK (2009/2010) and a Research Associate at College of Information Science and Technology, Drexel University, Philadelphia, USA (2004/2005). Dr. Xhafa holds a permanent position of Professor Titular at the Department of LSI, UPC (Spain). His research interests include parallel and distributed algorithms, combinatorial optimization, approximation and meta-heuristics, networking and distributed computing, Grid and P2P computing. Dr. Xhafa has widely published in peer reviewed international journals, conferences/workshops, book chapters and edited books and proceedings in the field. Dr. Xhafa has an extensive editorial and reviewing service. He is Editor in Chief of the International Journal of Space-based and Situated Computing, and International Journal of Grid and Utility Computing, Inderscience Pubs. Dr. Xhafa is actively participating in the organization of international conferences. His personal webpage can be found at http://www.lsi.upc.edu/~fatos/.

Florin Pop, PhD, is lecturer with the Computer Science Department of the University Politehnica of Bucharest. His research interests are oriented to: scheduling in Grid environments (his PhD research), distributed system, parallel computation, communication protocols and numerical methods. He received his PhD in Computer Science in 2008 with Magna cum laudae distinction. He is member of RoGrid consortium and participates in several research projects, in collaboration with other universities and research centers from Romania and from abroad developer.

Valentin Cristea is a professor of the Computer Science and Engineering Department of the University Politehnica of Bucharest (UPB). He teaches courses on Distributed Systems and

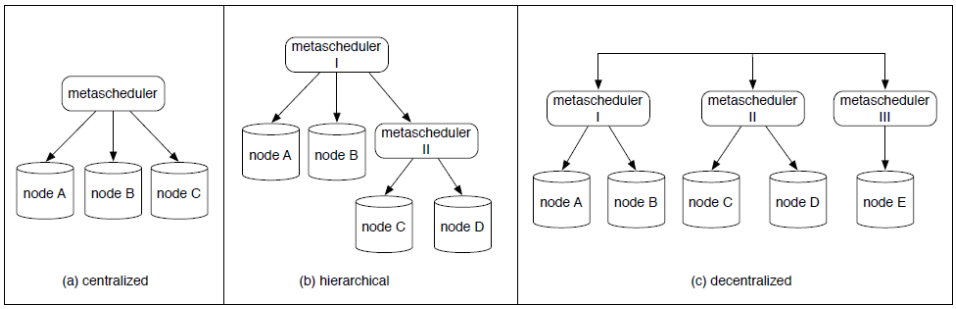
Algorithms. As a PhD supervisor he directs thesis on Grids and Distributed Computing. Valentin Cristea is Director of the National Center for Information Technology of UPB and leads the laboratories of Collaborative High Performance Computing and eBusiness. He is an IT Expert of the World Bank, Coordinator of national and international projects in IT, member of program committees of several IT Conferences (IWCC, ISDAS, ICT, etc), reviewer of ACM.

1 Introduction

In recent years, the clouds turn out to be one of the most promising computing paradigms that have emerged in terms of arisen expectation of distributed services including hardware and software (Chuang et al., 2011). This is because of clouds' great capacity to offer scalable and rigid elastic services, which is one of the most closely scrutinised research areas in wide-scale computing systems (Kim, 2011, Peiris et al., 2011). In such environments, the vast computing resources, which reside at a remote location, offer on-demand varied-priced flexible services, including hardware, software and developers' platforms. The elastic level on demands availability could be leveraged towards an improved quality of service. In general cloud computing scales resources of data-centre in a satisfactory level of the quality of service (Petruch et al., 2011 Gong et al., 2010) yet there is a concern that when end-users number and/or resource demand increase, the capacity-oriented clouds (e.g. data or storage clouds) will reach their maximum service equilibrium (Sasikala et al., 2011, Buyya et al., 2010, Sotiriadis et al., 2011). This will pose new capability-oriented needs for optimising the overall quality of services among multi-clouds (Bessis et al., 2011a, 2011b).

In this work, we take the view and define InterCloud of inter-collaborative and inter-cooperative enterprises as a temporal auto-scaling resource formation in which services and resource exchange happen among various clouds but also amongst other e-infrastructures as to augment service quality and provide a total satisfaction for a wide range of customer diverse requirements. Such e-infrastructures should include but not limited to clusters, grids, high performance and throughput computing. In general, the InterCloud approach expand the cloud capabilities in terms of services with the aim of achieving a wider distribution of resources, yet by retaining global resource utilisation equilibrium among various resource pools. In such settings, one of the most important design issues for an inter-collaborative cloud is the resource scheduling strategy with respect to its local cloud data-centre scheduling plan.

Figure 1 The meta-scheduling schemes



Specifically, the approach implies that a locally located resource could participate in the on-demand resource selection process at both local (intra-) and global (inter-) 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 as demonstrated in Figure 1. In previous works (Bessis et al., 2011a, 2011b, Sotiriadis et al., 2011) we have presented a comparative study of various schedulers. We have deliberated 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 InterCloud environments including scheduling, simulation and testing. The next sections present the motivation, the state-of-the-art review of the related approaches, the requirements analysis and the simulation, configuration and experimentation issues of the study.

2 Motivation

The cloud paradigm shares several commonalities with other technologies including grid and utility computing by combining their most important characteristics in order to offer a variety of services i.e. infrastructure as a service etc. (Kim, 2011). Clearly, the key features of these technologies could contribute towards interoperable infrastructures when additional complexity is added to clouds – in the case of InterCloud – by considering the meta-scheduling paradigms (Sotiriadis et al., 2012). For example, the grid characteristic of resource geographical distribution and the use of virtualisation technology in utility computing highlight new requirements for clouds. Similarly, in case of inter-enterprises of cloud-to-cloud and/or cloud-to-grid, also known as InterCloud, new requirements are required for achieving optimum scheduling performance.

In general, InterCloud could be considered as a wide research effort in which issues such as resource discovery, allocation and scheduling are quite the most important. In this work we only focus on the job scheduling aspect for distributed environments (grids and clouds) so deliberately, we aim to identify schedulers that are easy to adopt within InterCloud e-infrastructures. However, one of the most important criterion for our selection is the dynamics of a system as the unpredictability of resources is high and as described in Huang et al., 2011, Korkhov et al., 2009 and Wang et al., 2010. This is to say that dynamic-ness of a meta-scheduling approach forms the basis of our research. Specifically, static-ness in scheduling is defined as the approach in which all the decisions are done prior to the execution of the schedule. On the other hand, dynamic-ness allows decisions during the execution time. This enables the consideration of unforeseen situations that may occur in large-scale, fluid type of e-infrastructures. Thus, the generic problem area of the research study is motivated by the dynamically changed nature of an InterCloud and e-infrastructure (infrastructure from now on) environments.

During the past years scheduling within uncertain environments in terms of scale (e.g. grids) has been extensible studied. A great variety of scheduling algorithms (centralised or distributed) have been proposed by Andrade et al., 2003, aiming to a more flexible and efficient operation. However Buyya et al., 2010, suggest that when things come to developments and use-cases aiming at testing a realistic solution it turns to be impossible to design such tools. This is mainly because of important characteristics such as support for dynamic scheduling are not considered. In addition, the clear problem is that the actual requirements are not known in advance, which may cause a change of the initial conditions and chosen parameters. Thus, the strategy for developing such solutions should be fully automated, and flexible in terms of considering dynamic metrics as much as possible.

In this work, we identify those metrics on the basis of a literature study about various meta-scheduling approaches. In particular, those metrics that are closely related to the distributed meta-scheduling scheme. This is because the latter is concerned with the distribution of jobs across independent sites in distinct administrative areas. The decentralisation aims to overcome the common problems such as the single point of failure and bottleneck of the cloud environments caused by a central instance that has a solely responsibility for handling all jobs and request. Thus, we anticipate that the distributed meta-scheduling approach where various local resource management systems (LRMS) will deploy their own meta-scheduling strategies for job delegations and executions. We continue by discussing the most common meta-approaches for identifying the most crucial characteristics that lead to the fundamental requirements. Therefore, a discussion of scheduling topologies (centralised, hierarchical and decentralised) which are applicable for meta-computing settings is presented next. Specifically, by analysing their advantages and drawbacks the state-of-the-art aims to identify characteristics and commented research gaps towards an InterCloud set-up.

3 Related Works

In recent years, the InterCloud as a term has been emphasised by the leading vendors in cloud services area such as HP, Intel, Yahoo, etc. (Calheiros et al., 2011). It is noticeable that their state-of-the-art efforts have led to the establishment of a federation of collaborated clouds with joint initiatives. However, this vendor-oriented endeavour of InterCloud has a specific control plane rather than a setting that it is based on future standards and open interfaces which are available to be shared in the academic community. In addition, knowledge sharing, experimentation and testing within their systems have been limited to the wide range of researchers. In contrast to aforementioned work, the vision of InterCloud as an inter-cooperative infrastructure including inter-enterprises has been introduced by Buyya et al., 2011 yet from a federated perspective.

They suggest a utility-oriented federation of various cloud computing environments. They conclude to a business model of system architecture including the most important elements (requirements) of InterCloud in terms of complete system components. However, to the best of our knowledge, none of the aforementioned works deal directly with the meta-scheduling concept of interoperable clouds, thus the literature review emphasis is on the scheduling concept of jobs in high dynamic environments. The family of job scheduling problems has been extensively studied in distributed computing literature and thus, it is considered beyond of the scope of the study. The aim herein is on specific topologies that could be adapted in scheduling in InterCloud and share a common aim; addressing dynamic situations in large-scale collaborative settings (Bouafra et al.,2011). In the context of meta-scheduling three topologies are identified from the literature namely centralised, hierarchical and distributed meta-scheduling. Next, a discussion of each topology along with various schedulers is presented starting with the centralised and hierarchical scheme and concludes to the most challenging the distributed (also known as decentralised).

3.1 Centralised & hierarchical meta-scheduling

In the centralised model meta-scheduling happens directly by a central instance (Xhafa et al., 2010) which maintains information of all resources. Each time new jobs are submitted; the centralised meta-scheduler either sends the jobs for execution or in the case that execution cannot start, due to inaccessibility, arranges the jobs in a queue. Specifically, the resources do not perform scheduling decisions but only act as dispatchers while the local sites inform the meta-scheduler for job completion and availability of computational resources.

Starting from the late 1990s efforts in meta-computing mainly target the identification of the best possible scheduling algorithm. The great advantage of this method is that the central administration has a complete knowledge of the actual environment, so common concerns in scheduling such as starvation could be easily predicted. In addition, the meta-

scheduler assigns jobs constantly to the best possible resource for execution by selected jobs from a pool list vector. This is the main reason that centralised works claim to offer very good performance results (Shah et al., 2007). However, for each centralised meta-scheduler a local system administrator maintains the complete control, thus making systems' dynamic changes reasonable only in small scale settings. When the environment extends, issues like bottleneck and single point failure make them impractical. Having said that, high dynamics are very important to be overlooked (Huang et al., 2011). Consequently, the centralised scheduling scheme doesn't offer the full functionalities desired from an InterCloud scenario. This is because a more complex and sophisticated solution should be involved. The hierarchical meta-scheduling scheme is analogous to the aforementioned centralised scheduling. In this multi-administrative domain scheme, jobs are submitted to a central instance of the meta-scheduler which hierarchically communicates with other hosts that have their local schedulers with different policies. In general, this solution is considered as more advanced as compared to the centralised ones (Iosup et al., 2008).

In general, the hierarchical scheduling scheme has not been fully utilised by developers, mainly because it offers identical advantages and drawbacks to a centralised scheduler. To conclude, both approaches, centralised and decentralised, always offer remarkable results, and it could be a good practise to use them as a basis of comparison when developing highly dynamic distributed meta-schedulers for large scale environments. In the next section the distributed meta-scheduling topology and a variety of related scheduling approaches are discussed. With this in mind, the centralised and hierarchical scheduling scheme is considered as insufficient in serving the desired InterCloud scenario mainly because of their low capacity for handling interoperability and dynamics.

3.2 Distributed (Decentralised) meta-scheduling

The distributed meta-scheduling scheme 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 computational load data (Butt et al., 2003), and to find the site with the lowest load for transferring the job. This solution is the more advanced and complex as compared to centralised and hierarchical schemes as it is more scalable and flexible. Specifically, the meta-scheduler has a partial and instantaneous knowledge of the environment. This incomplete knowledge based solution is more realistic and usually related to granularity of the system. In contrast, centralised and hierarchical schedulers have a complete knowledge of the actual resource infrastructure. This includes the number of hosts, number of jobs submitted, the workload of each hosts, and the topology of the system.

In contrast, in the distributed scheme, this information is incomplete and the jobs received from the meta-scheduler are assigned to the local scheduler of the same or a different host. As all the jobs are submitted locally the distributed scheme allows jobs to be transferred to remote hosts for achieving better local resource utilisation, thus leading to global load equilibrium. The distributed or decentralised meta-scheduling approach could offer significant enhancement towards the InterCloud scheduling decision mainly because of its great ability with interoperability issues. In view of that, next we include a brief discussion of the most common distributed meta-scheduling algorithms that have been studied over the years in academia as presented in Sotiriadis et al., 2012. Starting with the work of Weissman et al., 1996, they propose a wide-area scheduling system based on a local resource management system (LRMS) and a wide-area scheduler. Each member of the site has to instantiate a) the LRMS which manage the local resources and b) the wide-area scheduler (WA) which achieves a global scheduling. In Anand et al., 1999 authors discuss a decentralised dynamic algorithm namely estimated load information scheduling algorithm. The method first estimates the load awaiting service (queue length) at the neighbourhood processors and secondly reschedules the loads at the current resource based on these estimates.

The work of Frerot et al., 2000 present a model namely federation of distributed resource traders and parallelise jobs submissions to user defined services. By coupling several resources to providers the resource trader acts on a similar fashion as to a meta-scheduler as the intermediary among consumers and providers. The authors of Subramani et al., 2011 demonstrate a distributed computing scheduling model which "adapts to changes in global resource usage" (Shah et al., 2007). The key idea of the meta-scheduler is to redundantly distribute each job to multiple sites, instead of sending the job to the most lightly loaded. In (Butt et al., 2003) authors present a model for connecting various Condor work pools which yields to a self-organising flock of Condors. This work is more focused in the area of resource discovery by using a P2P routing Pastry model.

Andrade, 2003, proposes a scheduling infrastructure based on the bag-of-tasks applications called OurGrid. The OurGrid is a collection of peers constituting a community. Lai et al., 2004, discusses a market-based resource allocation system in which the scheduling mechanism is based on auctions. Each resource provider or owner runs an auction for his resources. The work of Shah, 2007, suggests two scheduling algorithms namely the modified ELISA (or MELISA) and the load balancing on arrival. Both algorithms are based on the distributed scheme of sender-initiated load balancing. Their difference is in the grid scaling as MELISA works better in large scale systems, and load balancing on arrival works well with small scale environments. Iosup, 2008, discusses the delegated matchmaking (DMM) approach as a novel delegated technique which allows the interconnection of several grids without requiring the operation of central control point by temporarily bind local resources to remote resources.

In a similar vein, De Assuncao et al., 2008, present a model for the InterGrid as a sustainable system. The authors first discuss on existing research studies with the aim of

creating national and continental grids. Leal et al., 2009, present a decentralised model for addressing scheduling issues in federated grids. This solution proposes the utilisation of the GridWay; a meta-scheduler to each grid infrastructure of the federated grid. The method is an alternative to the centralised setting. The work of Rodero et al., 2010, discusses the problem of broker selection in multiple grid scenarios by describing and evaluating several scheduling techniques. In particular, a system entity e.g. hosts and grid virtual organisations are represented as meta-brokers which might behave as gateways. Wang et al., 2010, suggest the problem of overloading by suggesting an alternative mean of resource selection called bidding. They claim that there is no global information available in a dynamic environment e.g. grid and cloud, bidding cannot facilitate optimum decision.

The work of Huang et al., 2011, introduces a decentralised dynamic scheduling approach called community aware scheduling algorithm (CASA). The CASA functions as a two phase scheduling decision and contains a collection of sub-algorithms to facilitate job scheduling across decentralised distributed nodes Finally, Buyya et al., 2010 introduce a model for InterCloud by focusing on the generic architectural issues, including the broker and the coordinator of clouds. The actual scheduling algorithm is a simple time and space-shared solution. To conclude, this section presented the related scheduling approaches with the aim of identifying the key characteristics of the meta-scheduling. A detailed evaluation of selected approaches is presented in Sotiriadis et al., 2011. Next, we present the actual requirements as drawn from the literature.

4 Requirements analysis

The meta-scheduling scheme has proven to be a very promising approach because of its capability to handle efficiently scalability and flexibility issues in large scale resource pools. Various approaches have been discussed in section 4, and each one has been developed to address different requirements. Thus, it could be said that meta-scheduling schemes are classified according to their effectiveness in bridging the gap of resource amongst large scale and various size infrastructures. In our case – scheduling in InterCloud – centralised and hierarchical solutions are considered impractical for such settings. This is because issues like unique administration management, single point failure, and local resource management dependencies could lead to crucial complications for the whole environment. That is the reason because the majority of the meta-scheduling approaches have been developed in a decentralised fashion.

The following discussion summarises the most important characteristics that could lead to the identification of relevant requirements of this study. The specific characteristics are derived from the cross-evaluation of various works as presented in Bessis et al., 2011a, 2011b. Heterogeneous pool of resources is recognised as one of the crucial subjects in various cases e.g. Wang et al., 2010, Andrade et al., 2003. However, the literature study shows that tentative results from the aforementioned works confirm a

low appreciation of the heterogeneity issue during experimentation. The interoperability and flexibility between local and meta-schedulers is subject to the requirements posed by the desired scenario. In any case both issues are considered in various works by either supporting scheduling autonomy as in (Weissman et al., 1996), temporary binding amongst resources and jobs (Iosup et al., 2011). The dynamic-ness of the infrastructure environment is a critical property when developing an interoperable meta-scheduler. Various works attempt to solve meta-scheduling issues derived from the unpredictability of a dynamic changing environment as in Leal et al., 2009, which consider past performance requirements for forecasting new objectives. Similarly, Huang et al., 2011, present a meta-scheduling tactic that doesn't expose internal node information and based on nodes' real time responses.. In contrast with those solutions, non-dynamic approaches such as Iosup et al., 2011 and Leal et al., 2009 assume a steady-state setting during simulation. In the latter approach, authors suggest a delegated matchmaking procedure in which resources are matched temporarily to remote resources.

The geographical distribution of different pool of resources is considered in most of the works as they all include meta-scheduling for grid environments. Specifically, Frerot et al., 2000, present scheduling strategies for geographically distributed resource pools e.g. grid virtual organisations. Normally, this issue is part of the overall objective, the distributed meta-scheduling of jobs. Inter-collaboration for sharing resources and/or jobs amongst same and/or different infrastructures e.g. grid virtual organisations and HPC, grids and clouds is usually neglected as the complexity in such settings is exponentially rising mainly because of the additional requirements. Specifically, the works of De Assuncao et al., 2009 aiming to an inter-grid of interlinking grid collaborated islands using peering arrangements.

The rescheduling concept and advance reservation mechanism are commonly used in various cases for iteratively improve the performance of the scheduling process. Specifically, in Huang et al., 2011 authors claim that during a rescheduling phase a notable improvement has been observed in the scheduling performance. Equally, Leal et al., 2009, suggest that by utilising an advance reservation mechanisms based on previous works performance measures, a significant enhancement in performance has also being observed. However, the authors suggest that the overhead during training may be increased in terms of large scale job input. Finally, decoupling, on the other hand decides the delegation of jobs from site to site without connecting resources.. Using an evolutionary computation method optimise workload exchanging. Similarly, Rodero et al., 2010, presents a policy that considers dynamic performance metrics, based on backfilling uses dynamic performance information and finally Huang et al., 2011, performs scheduling of jobs based on dynamic real time node responses. To conclude, a detailed analysis on meta-scheduling algorithms has been discussed. The next section presents the simulation model of the interoperable approach, including different simulation prototypes of HPC, grid and cloud.

5 Simulation set-up

The section aims of identifying the simulation environments that fulfil the problem specification namely as dynamic job scheduling in collaborative Clouds. Specifically, an InterCloud meta-scheduler essentially requires a high dynamic, heterogeneous and decoupled simulation environment as discussed in previous section. In addition, the support of cloud features such as virtualisation is vital to the final design decision. The work includes the next discussion of most common simulators by presenting their advantages and drawbacks and their applicability to the specific scenario.

Several simulation environments have been developed over the recent years for offering advanced scheduling decisions. This alternative solution to the real-world scheduling systems allows researchers to evaluate the hypothesis prior to the software development and implementation process in order to test the actual behaviour of the system in several scenarios, metrics and criteria. However, different scheduling criteria and metrics are utilised from developers for running experiments for various scenarios. To this extend, in the case of this research, scheduling in InterCloud; it has been proposed by (Calheiros et al., 2011) that none of the conventional schemes as SimGrid (Casanova et al., 2001) and GridSim (Gallardo et al., 2009) which are mainly event-driven simulators, could address directly the cloud modelling requirements (e.g. support of virtualisation, dynamics, heterogeneity and loosely-couple-ness).

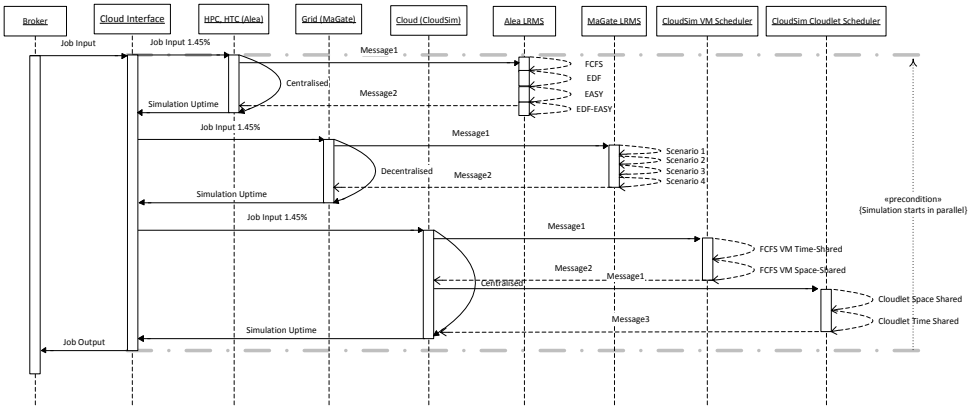
In parallel, the work in Klusacek et al., 2010, evaluate various simulation frameworks e.g. GangSim and SimGrid and conclude that although the aforesaid toolkits offer grid simulation capabilities, none of these could support directly the posed requirements (application and infrastructure) arising from cloud computing and especially for InterCloud. Accordingly, efficient simulation machine that offers dynamic scheduling decisions such as the Alea (Klusacek et al., 2010) SmartGRID (Huang et al., 2008), and the CloudSim (Calheiros et al., 2011) first incorporate and then extend conventional simulator schemes by integrating dynamics. For instance, Alea that is based on GridSim extends functionality for handling dynamic situations. In addition, SmartGrid is based on Alea and GridSim which both bring “the modelling of different kinds of essential grid components, such as grid jobs with various parameters, heterogeneous grid resources, and grid users” (Klusacek et al., 2010). Finally, CloudSim , originally bases its design in GridSim, however it includes important features for modelling and simulating large scale clouds data centres, virtualised servers, customised policies, energy-aware computational resources, federated clouds, user-defined policies and dynamic insertion of simulation elements. To conclude, our experiments will be performed in various simulation environments such as Alea, SmartGRID and CloudSim. This is because they offer appropriate simulation environments for the evaluation study of the proposed InterCloud scheduler.

5.1 The simulated architecture integration and algorithms

The integrated cooperative infrastructure consists of five components namely as the Broker who is responsible for inputting jobs within the system on behalf of a user, the Cloud Interface which is responsible for redirecting the same workload amount (in our case with input size 1.45%) to firstly a HPC or HTC infrastructure (simulated with the Alea), secondly to a Grid computing infrastructure (simulated with the MaGate) and thirdly in a Cloud system (simulated with the CloudSim).

Specifically, each simulation environment contains a number of local resource management systems (LRMS). For our experiment we utilise the first come first serve (FCFS), the earliest deadline first (EDF), the EASY Backfilling and the hybrid EASY Backfilling with EDF as LRMS in the Alea simulator. Similarly the MaGate implements four scenarios as discussed in the literature that could run in FCFS, Shortest Job First (SJF) and EASY backfilling schedulers. Finally, the CloudSim executes cloudlets (jobs) in twofold, firstly the scheduling decision is affected by the way in which VMs are allocated within hosts and secondly by the delegation of cloudlets in the VMs. Both cases are executed in FCFS fashion with either space-shared or time-shared scheduling tactic (Buyya et al., 2010).Figure 2 illustrates the integrated system that contains all the components of the system. The next section contains the integration of the scheduling pseudo-code for the Cloud Interface, the HPC and HTC, Grid and Cloud settings.

Figure 2 The cooperating infrastructure of HPC, Grids and Clouds



5.1.1 The Cloud Interface

The Cloud Interface scenario implies to a centralised scheduling case of redirecting the same job input in the three different schedulers. This includes various algorithms implemented in the simulation machine will allow the extraction of the primary data-set that it will be that base of experimentation.

1: Integrated Cloud Interface

Require: W_i : the initial jobs number of the workload archive
$Jobs_i$: the total number of jobs
$Wper_i$: the percentage of the workload
$Jobs_{peri}$: the jobs according to the experiment workload percentage
$Pool_i$: the resource pool (number of clusters and nodes)
$Message_{start}$: the start execution message
$Message_{results}$: the job delegated results come directly from the remote scheduler
$Conf_i$: the configuration file contains the experiment data (total PEs, memory etc.)
$Results_{Infrai}$: the results from each infrastructure
CloudInterface: the Cloud Interface

Require: HPC($Jobs_{peri}$, $Pool_i$, $Conf_i$): the HPC simulation setting (implemented with Alea)
Grid($Jobs_{peri}$, $Pool_i$, $Conf_i$):the Grid simulation setting (implemented with MaGate)
Cloud($Jobs_{peri}$, $Pool_i$, $Conf_i$): the Cloud simulation setting (implemented with CloudSim)
Compare($Results_{Infrai}$): the comparison of results for identifying best execution times

Require: Send message(), Get message()

1: **for** $jobs_i$ = {i, i++, n} W_i **do**
2: $Jobs_i$ * $Wper_i$ → $Jobs_{peri}$
3: **for all** $Jobs_{peri}$ **do**
4: Send Message($Jobs_{peri}$) **to** HPC($Jobs_{peri}$, $Pool_i$, $Conf_i$),
5: Grid($Jobs_{peri}$, $Pool_i$, $Conf_i$),
6: Cloud($Jobs_{peri}$, $Pool_i$, $Conf_i$)
7: Get $Message_{resultsHPC}$ $Message_{resultsGrid}$ $Message_{resultsCloud}$
8: **for all** $Message_{results}$ **do**
9: Compare($Results_{Infrai}$)
13: **end for**
16: **end for**

The above pseudo-code includes the functionality of the Cloud Interface. The last one is responsible for collecting data from the broker (who acts on behalf of a user) and redirects a message with a common configuration setting to each of the three simulators. Finally, the results return in an asynchronous sequence back to the interface that compares best execution times (for further evaluation) and send results (first come first send) back to the user through the broker.

5.1.2 The HPC Simulation

The HPC simulation is implemented within the Alea simulator and contains four LRMS. The following pseudo-code illustrates the functionality of the system when a job submission arrives in the Alea scheduler.

2: The HPC, HTC Scheduler (Alea)

Require: $Wper_i$: the percentage of the workload
$Jobs_{peri}$: the jobs according to the experiment workload percentage
$Pool_i$: the resource pool (number of clusters and nodes)
$Message_{start}$: the start execution message
$Message_{results}$: the job delegated results come directly from the remote scheduler
$Conf_i$: the configuration file contains the experiment data (total PEs, memory etc.)
$Results_{Alea}$: the results from each infrastructure
$Algorithm_i$: the LRMS algorithm (e.g. FCFS etc.)
Require: AlgorithmList{FCFS, EDF, EASY, EDF -EASY}: the algorithm list (four in our case)
StartSimulation ($Jobs_{peri,}$ $Conf_i$): the start simulation trigger
Get ExTime(): the total execution time of each job submission (scheduling uptime)
Get UtilisationPerc(): the percentage of utilisation level for each cluster (node)
Require: Send message(), Get message()

```
1:  Get message(Jobs_peri,, Conf_i)
2:  for Jobs_peri = {i, i++, n}  Wper_i do
3:          StartSimulation (Jobs_peri,, Conf_i)
4:          for all Algorithm_i ={i, i++, j}  AlgorithmList{FCFS, EDF, EASY, EDF -EASY} do
5:                  StartSimulation (Jobs_peri,, Conf_i)
6:                  Get ExTime(Algorithm_i)
7:                  Get UtilisationPerc(Algorithm_i)
8:                  Send message(Results_Alea, CloudInterface)
9:  end for
10: end for
```

The above pseudo-code includes the instantiation of the Alea simulator. The last one gets the configuration data from the Cloud Interface and starts the simulation. Finally, the results are send back to the Cloud Interface for interpretation.

5.1.3 The Grid Simulation

The Grid simulation is implemented within the MaGate simulator which contains three local scheduling algorithms (FCFS, SJF and EASY backfilling). In addition the MaGate implements four simulation scenarios to perform experiments in complete and partial knowledge domain. The following pseudo-code illustrates the functionality of the system when a job submission arrives in the MaGate scheduler.

3: The Grid Scheduler (MaGate)

Require: $Wper_i$: the percentage of the workload
$Jobs_{peri}$: the jobs according to the experiment workload percentage

```
                Pool_i: the resource pool (number of clusters and nodes)
                Message_start: the start execution message
                Message_results: the job delegated results come directly from the remote scheduler
                Conf_i: the configuration file contains the experiment data (total PEs, memory etc.)
                Results_Alea: the results from each infrastructure
                Scenario_i: the specific scenario
Require:        AlgorithmList{FCFS, EDF, EASY, EDF -EASY}: the algorithm list (four in our case)
                ScenarioList {S1, S2, S3, S4}: the scenarios list (four scenarios)
                StartSimulation (Jobs_peri,, Conf_i): the start simulation trigger
                Get ExTime(): the total execution time of each job submission (scheduling uptime)
                Get UtilisationPerc(): the percentage of utilisation level for each cluster (node)
Require:        Send message(), Get message()
1:              Get message(Jobs_peri,, Conf_i)
2:              for Jobs_peri = {i, i++, n}  Wper_i do
3:                      StartSimulation (Jobs_peri,, Conf_i)
4:                      for all Scenario_i ={i, i++, j}  ScenarioList {S1, S2, S3, S4} do
5:                              StartSimulation (Jobs_peri,, Conf_i)
6:                              for all Algorithm_i ={i, i++, j}  ScenarioList {S1, S2, S3, S4} do
5:                                      Get ExTime(Algorithm_i)
7:                                      Get UtilisationPerc(Algorithm_i)
8:                                      Send message(Results_Alea, CloudInterface)
9:              end for
10:  end for
```

The above pseudo-code includes the job delegation of a job submission within the MaGate simulator. After the completion, the results are sending back to the Cloud Interface for interpretation.

5.1.4 The Cloud Simulation

The cloud simulation is implemented within the CloudSim simulator which contains two local scheduling algorithms (FCFS in space and time shared fashion). In addition the Cloudsim implements two simulation scheduling cases for performing experiments in scheduling of VMs to Hosts and cloudlets to VMs. The following pseudo-code illustrates the functionality of the simulator.

4: The Cloud scheduler (CloudSim)

```
Require:        Wper_i: the percentage of the workload
                Jobs_peri: the jobs according to the experiment workload percentage
                Pool_i: the resource pool (number of clusters and nodes)
                Message_start: the start execution message
                Message_results: the job delegated results come directly from the remote scheduler
                Conf_i: the configuration file contains the experiment data (total PEs, memory etc.)
                Results_Alea: the results from each infrastructure
                VMScheduler: the VM to hosts scheduling algorithms
                VMSch_i: a specific the scheduling decision
                CloudletScheduler_i: the cloudlet to VM scheduler
```

```
          Algorithm_i: the LRMS algorithm (e.g. FCFS SS)
Require:  LRMS {FCFS SS, FCFS TS}: the LRMS scheduler (SS: Space, TS: Time shared)
          StartSimulation (Jobs_peri,, Conf_i): the start simulation trigger
          Get ExTime(): the total execution time of each job submission (scheduling uptime)
          Get UtilisationPerc(): the percentage of utilisation level for each cluster (node)
Require:  Send message(), Get message()
1:        Get message(Jobs_peri,, Conf_i)
2:        for Jobs_peri = {i, i++, n}  Wper_i do
3:                StartSimulation (Jobs_peri,, Conf_i)
4:                for all VMSch_i ={i, i++, j}  VMScheduler do
5:                        StartSimulation (Jobs_peri,, Conf_i)
6:                        for all CloudletScheduler_i ={i, i++, j}  LRMS {FCFS SS, FCFS TS} do
5:                                Get ExTime(Algorithm_i)
7:                                Get UtilisationPerc(Algorithm_i)
8:                                Send message(Results_Alea, CloudInterface)
9:        end for
10:  end for
```

To conclude the above pseudo-code includes the job delegation of a job submission within the CloudSim simulator. After the completion, the results are sending back to the Cloud Interface for interpretation.

5.2 Experimental results

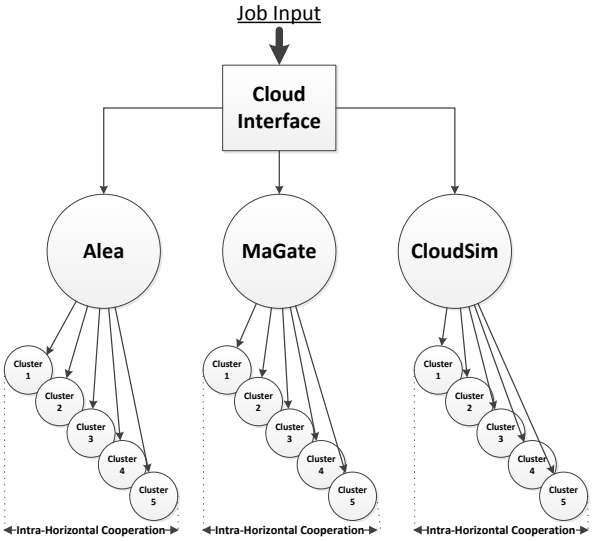
In this work, we utilise three simulation frameworks namely as Alea, MaGate and CloudSim in order to perform experiments based on real job workload archives and synthetic. Specifically, for the cases of Alea and Magate we use real workload traces from the Grid Workload Archives (GWA website) and for doing experiments in CloudSim we use a synthetic configuration identical to the real workload. All three simulation environments allow experiments in distributed systems; however each one has been developed for different purpose.

Alea utilises a centralised queue-based advanced scheduling technique for scheduling jobs in grid and cluster-oriented settings. MaGate, is a simulator supporting real grid workload traces by incorporating a dynamic scheduling algorithm implemented within four different static and dynamic scenarios. Finally, CloudSim is a simulation environment that offers Cloud embedded functionalities e.g. generation of datacentres, virtual machines (VM), and hosts. In addition, CloudSim has a significant difference with both aforementioned frameworks by supporting two fold scheduling, firstly the way in which VMs are orchestrated by hosts, and secondly the cloudlets (jobs) queuing of user demands within the VMs of host.

The aim of this experiment is to test the performance of each environment in terms of execution time of certain job traces based on the fact that each scheduling simulator has been implemented for different environments. From the perspective of scheduling, Alea initially simulates centralised scheduling for the case of a small scale infrastructure of

collaborated nodes of grid clusters in which each node has a complete knowledge of their domain. Secondly, MaGate simulates a large scale scheduling that uses nodes real time responses during the scheduling by incorporating static, dynamic and re-scheduling scenarios. At last, CloudSim simulates jobs submitted to a cloud datacentre and executed by virtual machines which are instantiated within the cloud hosts. This setting, by default, includes scheduling in two dimensions; the space shared (by assigning CPU cores to specific VMS) and time shared (dynamically distribute the capacity of a core among VMs) in first come first served fashion.

Figure 3 Cloud cooperative setting

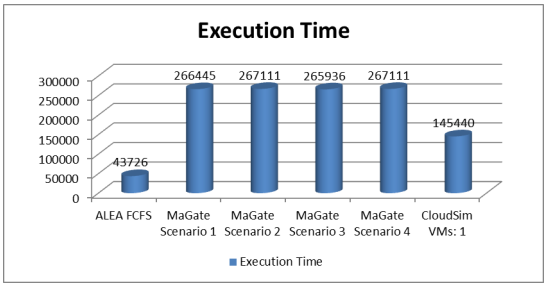


For measuring the performance of each simulation setting the study uses as a comparison metric the execution time of jobs (execution time in Alea, scheduling uptime in MaGate and scheduling time in CloudSim) by utilising the scheduling algorithms as implemented within the simulator. Alea includes various algorithms e.g. the FCFS, easy backfilling, earliest deadline first and the hybrid earliest deadline first with backfilling. Equally, for the MaGate scheduler, the community aware scheduling algorithm is tested in the FCFS, shortest job first and easy backfilling scheduling algorithms. In addition the CASA implements four different scenarios, in the first one each node in the pool receives the same amount of jobs which are eventually handled by the LRMS of each node, in the second centralised scenario the jobs are submitted to a unique centralised meta-scheduler which has a detailed knowledge of total PEs and PEs at any given time of each remote node. In the third decentralised scenario each node adopts a complete knowledge of the

remote nodes. Also, the nodes don't have a detailed knowledge of PEs of other nodes, so the CASA will be evaluated for job delegation behaviours. Fourthly, the decentralised scenario with partial knowledge of the resource discovery service each node has knowledge of certain remote nodes for delegating jobs (in the case of the AuverGrid input file 2 of total 5 clusters). Finally, the space-shared and time-shared in FCFS fashion within multiple cores are utilised for scheduling decisions in the CloudSim framework. Before discussing the further actions of the actual experiment, it is essential to present the static schedulers (e.g. FCFS) which their target is to assign tasks (processes or gridlets or cloudlets) to CPU cycles through a ready queue.

Having discussed that, we conclude that the assumption is to compare the performance of each setting based on their default schedulers by evaluating their execution and simulation times when the experiment is based on the same configuration input job file of a specific grid workloads archives namely as AuverGrid, 2011. This is a production grid platform consisting of 5 clusters in the region of France with a total value of 456 processing elements (PEs). Our target aim is to measure each environment (Alea: HPC, and cluster based, MaGate: Decentralised Grid, CloudSim: Cloud simulation platform) simulation when the same job input enters within a common cloud interface as described in Figure 3. It should be mentioned that each simulator at this time has the same configuration (5 clusters of total 456 PEs).

Figure 4 Total job execution times for different algorithms



The experiment uses 6000 jobs of total 404.176 (1.5%) of the AuverGrid workload trace file for each of the simulation setting. In addition, within each environment a horizontal cooperation (intra) happens by allowing jobs to run in different clusters. Table 1 presents the scheduling algorithms and their execution times for the case of FCFS job queues.

Table 1 Scheduling algorithms and their execution times for AuverGrid job input of 1.45%

Scheduling Algorithm	Total job execution time
ALEA FCFS	43726
MaGate Scenario 1	266445
MaGate Scenario 2	26711
MaGate Scenario 3	266445
MaGate Scenario 4	26711
CloudSim VMs: 1	145440

As illustrated in the table 1 the best job execution time is for the Alea FCFS scheduling queue (43726 ms.) while the MaGate scenario 1 and 3 have the worst time. Figure 4 illustrates the total job execution times from table 1. Based on the above values the following mathematic formulae (1) demonstrates the analogy of job execution for the configuration of 5 clusters of total 476 PEs when 1.45% of the AuverGrid grid workload archive enter in the interoperable cloud interface. Specifically, in the function the denotes the simulation environment (xa for the Alea, xβ for the MaGate, and xγ for CloudSim) while the i represents the number of jobs to be run within its. The ca, cβ, and cγ denote the coefficient values for the Alea for the MaGate for CloudSim respectively (the total job input is same.).

$$f(x_i) = 7.29 * x_a * c_a \cdot 44.51 * x_\beta * c_\beta \cdot 24.24 * x_\gamma * c_\gamma \qquad (1)$$

As derived from (1) the next formulae (2) illustrates a generic function of the workload for a different input workload archive (e.g. the Grid5000) where pi is given from the mathematic formulae (3) and .is the coefficient value of the workload percentage.

$$f(x_i) = p_a * x_a * c_i \cdot p_\beta * x_\beta * c_i \cdot p_\gamma * x_\gamma * c_i \qquad (2)$$

$$p_i = executionTimePerLRMS_i \Big/ \sum_0^i Workloadc_i \qquad (3)$$

To conclude, this section presents a simulation configuration that demonstrates the collaboration of different infrastructures and their simulation environments. It is apparent from formulae (1) that the Alea simulator offers the best job execution results, when all nodes (5 clusters) are utilised. Similarly, the CloudSim simulator also presents some good results by utilising only one VM per node. In contrast, MaGate scheduler does not offer exceptional results because each scenario is implemented with the assumption that 2 out of 5 total nodes accept jobs concurrently. Also, it incorporates the partial knowledge functionality and the real time responses of nodes in contrast with Alea and CloudSim.

Table 2 demonstrates the best execution times when different algorithms have been implemented when the same job input (1.45%) enters the common Cloud interface.

Table 2 Scheduling algorithms and their best execution times for AuverGrid job input of 1.45%

Scheduling Algorithm	Execution Time
ALEA FCFS	43380
MaGate EASY Scenario 4	267110
CloudSim VMs: 50	1296
CloudSim 5 DCs 10 VMs:	29160

Figure 5 Total job execution times for different algorithms and CloudSim settings

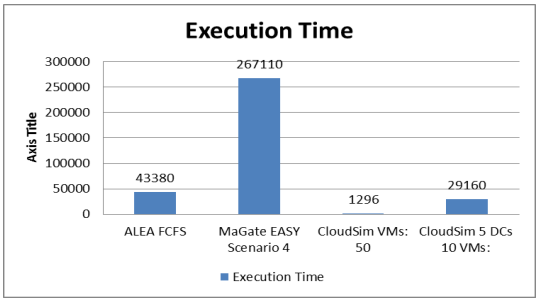
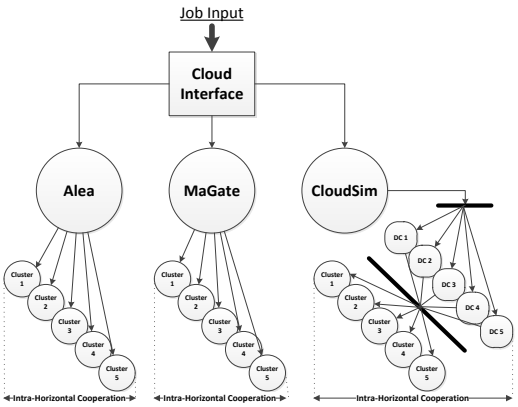
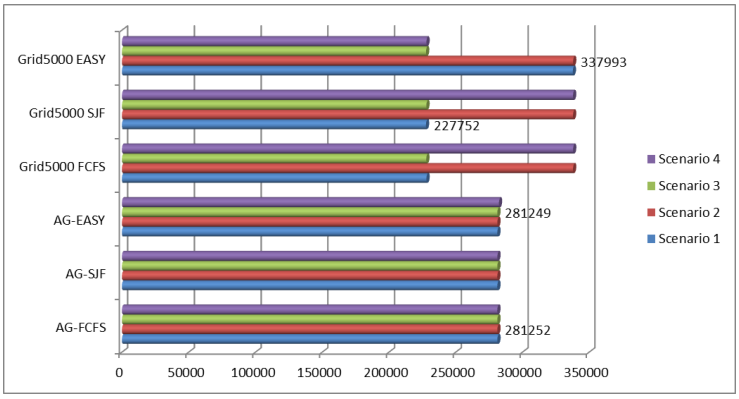


Figure 6 Cloud cooperative setting in AuverGrid input (1.45% workload)



Specifically, the Alea uses the FCFS, MaGate the EASY backfilling, and CloudSim uses the FCFS space-shares in two cases. The first experiment generates 50 VMs per node, while the second one contains 5 datacentres which generate 10 VMs per host. Figure 5 illustrates the aforementioned specification environment. In addition, figure 6 illustrates the cloud cooperative setting based on the cloud environment of 5 data centers.

Figure 7 MaGate performance in AuverGrid and Grid5000 traces



Finally, for illustrating the behaviour of the MaGate scheduler in large scale and heterogeneous settings we compare the simulation results when the FCFS scheduling execute jobs in the AuverGrid traces (404.176 jobs with 10% load) (GWA web site) configuration and the Grid5000 (1.020.195 jobs with 10% load) (GWA web site). The last one is an experimental geographically distributed grid platform consisting of 9 sites which each one is comprised by one or several clusters, for a total of 15 clusters (including heterogeneous machines) inside Grid5000. As we can see from figure 7, the MaGate improves the real-time scheduling significantly for the SJF algorithm and when the system is heterogeneous and extended in size as happened with the Grid5000 dataset.

6 Conclusion and future work

This study has been focused on the InterCloud meta-scheduling by presenting an analysis of the requirements and the future prospect for developing a novel meta-scheduler. Specifically, this work includes a state-of-the-art review towards the InterCloud scheduling including a discussion on the relevant technologies. By providing an overview of the problem area, the work presents the generic characteristics of cloud systems and the requirements for InterCloud as extracted from the literature study.

The study suggests three simulation environments for performing experiments, and results demonstrate the performance of a job input in a collaborative cloud interface. The future research step is to develop an inter-cooperative infrastructure in which jobs submitted within the cloud interface could be able to run in parallel within different environments based on the fastest execution time of the whole job input. In addition, the prototype will be expanded in order to support this functionality by implementing more complex algorithms in the CloudSim prototype.

7 Acknowledgements

The first author would like to thank UNITE FP7- 248583 for their secondment support. All authors would like also to thank the grid workload archives for the source of the workload archive and the AuverGrid and Grid5000 teams.

References

Anand, L., Ghose, D. and Mani, V. (1999) 'ELISA: an estimated load information scheduling algorithm for distributed computing systems'. *Computers & Mathematics with Applications*, pp. 57-85

Andrade, N., Cirne, W., Brasileiro, F. and Roisenberg, P., (2003) 'OurGrid: An approach to easily assemble grids with equitable resource sharing'. JSSPP'03: Proceedings of the 9th Workshop on Job Scheduling Strategies for Parallel Processing. LNCS, Springer, Berlin/Heidelberg, Germany

Bessis, N., Sotiriadis, S., Cristea V. and Pop, F. (2012) 'An architectural strategy for meta-scheduling in InterCloud', Proceedings of first International Workshop on InterCloud and Collective Intelligence (iCCI-2012), in conjunction with the 26th IEEE International Conference on Advanced Information Networking and Applications (AINA 2012). Fukuoka, Japan, March 26-29 2012 {To appear}

Bessis, N., Sotiriadis, S., Cristea V. and Pop, F. (2011)'Modelling Requirements for Enabling Meta-Scheduling in InterCloud and Inter-Enterprises', Proceedings of the. 3rd International Conference on Intelligent Networking and Collaborative Systems (INCoS 2011), IEEE CSP

Bouarfa, H. and Abed, M. (2010) 'The management of virtual collaborative design', Int. J.of Networking and Virtual Organisations(IJNVO), Vol. 7, No.1, pp. 99 - 107

Butt, A. R., Zhang, R. and Hu, Y. C. (2003). 'A self-organizing flock of condors', SC '03: Proceedings of the 2003 ACM/IEEE conference on Supercomputing. IEEE Computer Society, Los Alamitos, CA, USA

Buyya, R., Ranjan, R. and Calheiros, R. N. (2010) 'InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services', Algorithms and

Architectures for Parallel Processing (2010), Vol. 6081/2010, Issue: LNCS 6081, Springer, pp.13-31

Calheiros, R., N., Ranjan, R., Beloglazov, A., De Rose, A. F., C. and Buyya, R. (2011) 'CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms', Softw. Pract. Exper, 41, 1 , pp.23-50

Casanova, H. (2001). 'Simgrid: A Toolkit for the Simulation of Application Scheduling', *Proceedings of the 1st International Symposium on Cluster Computing and the Grid* (CCGRID '01), IEEE Computer Society, Washington, DC, USA, 430-.

Chuang, C-C. and Kao, S-J., (2011) 'Adjustable flooding-based discovery with multiple QoSs for cloud services acquisition', Int. J. Web and Grid Service, Vol. 7, No.2 pp. 208 - 224

De Assuncao, M. and Buyya, R. (2009) 'Performance analysis of allocation policies for interGrid resource provisioning', Information and Software Technology, pp.42-55

Frerot, C. D., Lacroix, M. and Guyennet, H. (2000) 'Federation of resource traders in objects-oriented distributed systems'. PARELEC'00 August 27 - 30, Quebec, Canada

Gallardo, A., Cerio, L. D., Messeguer, R., Isern-Deya, A. P., and Sanjeevan. K., 2009

Gong, C., Liu, J., Zhang, Q., Chen, H. and Gong, Z. (2010) 'The Characteristics of Cloud Computing', 2010 39th International Conference on Parallel Processing Workshops (ICPPW), 1316pp.275-279

Huang, Y., Bessis, N., Norrington, P., Kuonen, P. and Hirsbrunner, B. (2011) 'Exploring decentralized dynamic scheduling for grids and clouds using the community-aware scheduling algorithm', Future Generation Computer Systems, In Press, Accepted Manuscript, Available online 13 May 2011, ISSN 0167-739X

Huang, Y., Brocco, A., Kuonen, P., Courant, M. and Hirsbrunner, B., (2008). 'SmartGRID: A Fully Decentralized Grid Scheduling Framework Supported by Swarm Intelligence'. *Proceedings of the 2008 Seventh International Conference on Grid and Cooperative Computing* (GCC '08). IEEE Computer Society, Washington, DC, USA, pp.160-168

Iosup, A., Tannenbaum, T., Farrellee, M., Epema, D. and Livny, M., 'Inter-operating grids through delegated matchmaking', Scientific Programming 16 (2) (2008) pp.233-253

Kim, W. (2011) 'Cloud Computing Adoption', Int. J. Web and Grid Service, Vol.7, No.3, pp. 225 - 245

Klusacek, D., and Rudova, H.(2010). 'Alea 2: job scheduling simulator' *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques* (SIMUTools '10). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, Article 61.

Korkhov, V. V., Moscicki, J. T. and Krzhizhanovskaya, V., V. (2009) 'Dynamic workload balancing of parallel applications with user-level scheduling on the Grid', Future Generation Computer Systems, Vol.25, Issue 1, January 2009, pp.8-34, ISSN 0167-739X

Lai, K., Huberman, B. A. and Fine, L. (2004) 'Tycoon: an Implementation of a Distributed market-based resource allocation system', Technical Report, HP Labs

Leal, K., Huedo, E. and Llorente, I.M. (2009) 'A decentralized model for scheduling independent tasks in federated grids', Future Generation Computer Systems, pp.840-852,21, 27

Peiris, C., Sharma, D. and Balachandran, B. (2011) 'C2TP: a service model for cloud', Int. J. Cloud Computing 2011 Vol.1, No.1, pp. 3 - 22

Petruch, K., Stantchev, V. and Tamm, G. (2011) 'A survey on IT-governance aspects of cloud computing', Int. J. Web and Grid Service, Vol.7, No.3, pp.268 - 303

Rodero, I., Guim, F, Corbalan, J, Fong, L. and Sadjadi, S.M. (2010) 'Grid broker selection strategies using aggregated resource information',Future Generation Computer Systems, pp.72-86, 2010. 21, 26

Sasikala, P. (2011) 'Cloud computing: present status and future implications', Int. J. Cloud Computing 2011, Vol.1, No.1 pp. 23 - 36

Shah, R.,Veeravalli, B. and Misra. M. (2007) 'On the design of adaptive and decentralized load balancing algorithms with load estimation for computational grid environments', IEEE Transactions on parallel and distributed systems, pp.1675-1686

Sotiriadis, A., Bessis N. and Antonopoulos, N. (2012) 'From meta-computing to interoperable infrastructures: A literature review of schedulers and simulators', Conference on Advanced Information Networking and Applications (AINA 2012) , Fukuoka, Japan, March 26-29 2012 {to appear }

Sotiriadis, S., Bessis N. and Antonopoulos, N. (2012) 'Towards InterCloud schedulers: A survey of meta-scheduling approaches', Proceedings of 6th International Conference on P2P, Parallel, Grid, cloud and Internet Computing (3PGCIC 2011), Barcelona, Spain, Oct 26-28 2011

Subramani, V., Kettimuthu, R., Srinivasan, S. and Sadayappan, P. (2002). 'Distributed job scheduling on computational grids using multiple simultaneous requests',11th IEEE International Symposium on High Performance Distributed Computing (HPDC-11), 23-26 July. IEEE Computer Society, Los Alamitos, CA, USA

The Grid Workload Archive (2011), http://gwa.ewi.tudelft.nl/pmwiki/, (retrieved 10/11/2011)

The AuverGrid team (2011), http://www.auvergrid.fr/, (retrieved 16/11/2011)

The Grid500 team (2011), http://www.grid5000.org/, (retrieved 16/11/2011)

Wang, C.M., Chen, H.M., Hsu, C.C. and Lee, J. (2010) 'Dynamic resource selection heuristics for a non-reserved bidding-based Grid environment', Future Generation Computer Systems, pp.183-197

Weissman, J. B. and Grimshaw, A. (1996), 'Federated model for scheduling in widearea systems', HPDC'96: Proceedings of the Fifth IEEE International Symposium on High Performance Distributed Computing.pp.542-550, August

Xhafa, F. and Abraham, A. 'Computational models and heuristic methods for Grid scheduling problems', Future Generation Computer Systems, Vol. 26, Issue 4, April 2010, pp. 608-621, ISSN 0167-739X