2012 26th IEEE International Conference on Advanced Information Networking and Applications

From meta-computing to interoperable infrastructures: A review of meta-schedulers for HPC, grid and cloud

[1]Stelios Sotiriadis, [1]Nik Bessis, [2]Fatos Xhafa [1]Nick Antonopoulos
[1]School of Computing & Maths, University of Derby, Derby, United Kingdom
[2]Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, Spain
[1](s.sotiriadis, n.bessis, n.antonopoulos)@derby.ac.uk, [2]fatos@lsi.upc.edu

***Abstract* — Over the last decades, the cooperation amongst different resources that belong to various environments has been arisen as one of the most important research topic. This is mainly because of the different requirements, in terms of jobs' preferences that have been posed by different resource providers as the most efficient way to coordinate large scale settings like grids and clouds. However, the commonality of the complexity of the architectures (e.g. in heterogeneity issues) and the targets that each paradigm aims to achieve (e.g. flexibility) remains the same. This is to efficiently orchestrate resources and user demands in a distributed computing fashion by bridging the gap among local and remote participants. At a first glance, this is directly 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. In addition, the term meta-computing, hence meta-scheduling, offers additional functionalities in the area of interoperable resource management because of its great proficiency to handle sudden variations and dynamic situations in user demands by bridging the gap among local and remote participants. This work presents a review on scheduling in high performance, grid and cloud computing infrastructures. We conclude by analysing most important characteristics towards inter-cooperated infrastructures.**

Keywords: Meta-scheduling, Grids, Clouds, Inter-enterprises, Inter-clouds

I. INTRODUCTION

During, the last decades, several large scale computing architectures have been emerged with regards to their operating case scenario. The most common of these – in terms of distribution of resources' workloads – are the high performance or high throughput computing, the grid and the cloud computing. Firstly, the high performance computing (HPC) is an owner centric resource provisioning architecture in which resources are locally owned, and clients have private access to the owner organisation [33]. The aim of HPC is to gain great computational power for solving complex problems [24], normally in a particular administrated environment. Secondly, in grid paradigm resources are locally and/or externally owned, thus a wider administration resource domain is observed. Members of the grid constitute a virtual organisation (VO) and could access to resources in a public manner [33]. In this case, heterogeneous resources may enter and leave the grid dynamically, while at the same time their capacity and performance might be altered. This makes the administration and scheduling a challenging issue. Thirdly, in cloud computing, resources can also be externally or internally owned forming a public, private or hybrid setting. The pay-on-demand cloud model allows users to access bespoke resources, which size is dynamically growing by utilising the virtualisation technologies [10]. This kind of dynamic sizing allows the cloud to create, migrate and auto-scale resources dynamically [33].

Each one of the aforementioned computing technologies has several advantages and drawbacks which have been studied by literature in detail and that is beyond the purpose of this study. However, an initial appreciation of their important characteristics could be the means to identify their most common issues towards meta-scheduling solutions [4], [5]. This solution will allow resources to contain a meta-component which is placed on the top of the local resource and is responsible for interoperable coordination with remote participants. It is apparent that the most stable solution is the HPC paradigm in which the capability to manage workloads is static and high. However, grid and cloud can offer high capacity with average capability, as well as supporting interoperability and heterogeneity concerns. To this extend the next section presents a discussion of various scheduling topologies [39] (centralised, hierarchical and decentralised) which are applicable for different computing architectures e.g. HPC, grids and clouds.

II. THE META-SCHEDULING REVIEW STUDY

A meta-scheduler is a term found frequently in the grid computing, which purpose is of establishing a wide policy control among resources. This includes negotiation and management of a pool of resources bounded to different administrative domains; the grid VOs. Their key functionality is to form the communication bus along the local level scheduler(s) for (re)directing user defined tasks to the best resource(s) for delegation. Best resources could be considered by several criteria that aim to the best performance in terms of computational power and execution time. Thus, by filling the gap of resource sharing within each local administrative domain, meta-schedulers focus towards to an interoperable, efficient and flexible environment.

At all times scheduling in meta-computing was a challenging area for researchers mainly due to new additional requirements [9], [13], [14], [32] posed by the

1550-445X/12 $26.00 © 2012 IEEE
DOI 10.1109/AINA.2012.15

IEEE computer society

promising innovative technologies (e.g. cloud, utility computing). One of the most important concerns is always the fact that different ownership of resources lead to different topologies. Thus, the architectural issue involves the needs to evolve to a more open setting to reach a better scaling of resources. In general these topologies, as discussed in [39] are classified initially by [25] into centralised, hierarchical, and decentralised scheduling. The following section presents the topologies of meta-schedulers as derived from the review study.

1) Centralised meta-scheduling

In the centralised model meta-scheduling happens directly by a central instance which maintains information of all resources [41]. Each time new jobs are submitted; the centralised meta-scheduler either sends the jobs for execution – or in the case that execution cannot start, as there is no availability – arranges the jobs in a queue. Specifically, the centralised meta-schedulers do not perform scheduling decisions [41] but only act as dispatchers. Finally, the local sites inform the meta-scheduler for job completion and availability of computational resources.

Starting from late 1990s efforts in meta-computing mainly target to find the best possible scheduling algorithm. Since the scheduling mechanism has been characterised as an NP-hard problem to solve in the most cases it is common to use heuristic algorithms for selecting groups of candidate resources. Initially, work in [11] discusses the scheduling problem of tasks that can be executed by multiple processors. Using this method a variety of theoretical algorithms and formulas for independent task scheduling have been developed, still without considering heterogeneity of resources. The majority of these algorithms are very difficult to be implemented in real cases [6] however they guarantee a good worst case scenario. In contrast, [20] discusses a scheduling system that gains benefits when "the scheduler considers both the computer availability and the performance of each task on each computer". This framework allows jobs to be executed in a distributed fashion, yet, in a centralised clustering meta-algorithm.

In a similar vein authors in [6] study the meta-scheduling issue by considering strategies which are not so complex to be implemented. A decision metric herein is that real-workloads could demonstrate good performance. The aim in this work is to develop a centralised approach for providing solutions for the Northrhine-Westphalian (NRW) meta-computer [6], a country-wide meta-computer located in Germany. The results may be promising but there are limited as more powerful algorithms have not been evaluated. Authors in [40] present an advanced reservation based meta-scheduling system. The fluctuation of the dynamic environment makes prediction of hosts' behaviours to become unpredictable. The estimation of execution times is varying depending on the local resource availability and usage, as well on traffic and communication patterns [42]. Specifically, the approach allows the local scheduler to select resources for job execution by assigning resources to meta-jobs usage. The scheduler called Ursala, utilises the Maui system scheduler that is capable of supporting multiple scheduling policies, dynamic priorities, reservations, and fairshare capabilities [28] for requesting resource availability and job information (cost and times). This centralised solution allows fully optimisation of scheduling, by minimizing the negative impact of already scheduled workloads [40]. Experimental results show an improvement of response times of supercomputer centres. Yet, in realistic case, the major drawback of this centralised approach relies on the fact that not all local schedulers provide advanced reservation support [42]. Scheduling for centralised supercomputers has been discussed also by [7]. Their work presents a statically mapping of meta-tasks (meta-jobs) to resources in a predictive manner for minimizing the total execution time of the meta-task. Specifically, they define as mapping the matching and scheduling procedure and they suggest that the problem of optimally mapping is a NP-hard problem. By comparing eleven different heuristic algorithms they provide a discussion that reveals which heuristic is capable of utilising for which scenario. Obtaining such knowledge from a static environment it will be useful in the case of applying heuristics for different scenarios as this can become the starting point.

The backfilling algorithm is also presented as a solution for centralised meta-scheduling in [45]. The authors design a global backfilling scheduler which tries to find gaps at clusters for queued jobs to other clusters. The algorithm is evaluated by using real workloads trace driven simulations, and the results show that the policy outperforms the independent site execution. Dissimilar to the above strategy, a placeholder monitoring and throttling algorithm has been introduced in [34], that works across distributed and local schedulers. The basic idea is to centralise the jobs of the workload into a meta-queue, then use the placeholder to move the job the next accessible queue and use late binding to offer flexibility. Authors in [24] discuss a self-scheduling policy for high performance computing. The algorithm works on a two level architecture. At the top, the job is scheduled by the dispatcher to a resource, and then the resource local scheduler schedules the job.

The meta-brokering concept has been introduced by [29] with the aim of easing the addition and usage of different resource brokers. To achieve it, a brokering portal has been designed to reach resources of different grids in an automated way. The portal or meta-broker is a scheduler standing on top of the local meta-scheduler. The evaluation results show significant improvements when compared to the conventional meta-scheduling systems. In [30] a meta-brokering approach is discussed, which in contrast with [29] supports grid interoperability. Specifically, the meta-broker sits on the top of a resource broker and uses meta-data to decide where to send the job. Such scheduling methods are called meta-brokering [32]. The solution "creates a meta-level above current resource management solutions by using technologies from the area of the semantic web" [32].

However, this is a centralised solution inappropriate to complex and dynamic systems such as grids and clouds. The work of [23] utilises self-centred agents to achieve a total load balancing infrastructure. Specifically, the Algorithmic Mechanism Design (AMD) theory [16] is utilised as the specification of payments to agents in a way that results in an

environmental equilibrium [16]. The work is based on a centralised model in which the local dispatcher decided the allocation and payments. Finally, the work concludes to a protocol that implements the mechanism. At last, the Bellagio system [3] contains a market base resource allocation system for federated distributed infrastructures. The whole procedure is coordinated by a centralised auctioneer who controls the bids for resources. Typically a bid includes the required resources, the computational processing duration, and an amount of virtual currency.

To conclude, all the aforementioned works aim to a centralised meta-scheduling environment in which a central component is responsible for the management of various local schedulers. The great advantage of the centralised topology is that through central administration it is achieved 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 the centralised pool list. This is the mainly reason that the above works claim to have very good performance results. However, for each centralised meta-scheduler a local system administrator maintains the complete control, thus making systems' dynamic changes unpredictable. In addition, possible situations such as bottleneck in responses and centralised failure are very important to be overcome. Next, the hierarchical scheme is presented along with different scheduling approach.

2) Hierarchical meta-scheduling

The hierarchical meta-scheduling scheme is similar to the aforementioned centralised scheduling. In this setting jobs are submitted to a central instance of the scheduler which communicated with other schedulers belonging to its hierarchy. An advanced solution of hierarchical scheduling has been presented in [7] as a geographically distributed HPC setting. Its architecture is based on three layers structure namely the computer centre software (CCS) for scheduling system, the resource and service description for specifying hardware and software components, and the service coordination layer for brokering and registering applications. This approach offers a modular and autonomous solution on each layer. It is also reliable and scalable as it is hierarchically organised in autonomously "CSS islands" and performs scheduling in space-sharing way using deadlines. The authors compare CSS with the Globus meta-computing directory service (MDS) [13], in a theoretical way. A noteworthy to mention difference is that Globus sees the environment as a huge virtualised meta-computer in contrast with CCS which resources may be distributed but must be accessible in one domain.

A framework called Sharp for secure distributed resource management is discussed in [21]. The system is based on the barter economy in which exchange must be made using a cryptographically signed object called Resource Ticket (RT). In Sharp, each site runs local schedulers for physical resources. The system is controlled by three entities; the site authority which maintains the hard state of the resource, the service manager (resource consumers), and the agents. The agents behave as intermediary among site authorities and resource consumers. To conclude, the hierarchical scheduling scheme has not been fully utilised by developers, mainly because it behaves similar to a distributed meta-scheduler, thus inherits drawbacks from the centralised scheduling. This is underlined by [17] who suggest that this solution is more centralised than decentralised as there is one central scheduling instance in which jobs are submitted within a hierarchy. In general, both approaches, centralised and decentralised, always offer remarkable results, and it could be a good practise to use them as basis of comparison when developing highly dynamic distributed meta-schedulers for large scale environments. Besides, various scheduling approaches compare e.g. [32] compare their results with a centralised and/or hierarchical solution to present their performance results. In the next section the distributed meta-scheduling topology and a variety of related scheduling approaches is discussed.

3) Distributed meta-scheduling

The distributed meta-scheduling theme 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 [12], and to find the site with the lowest load for transferring the job. This solution is the more advanced and complex, comparing with centralised and hierarchical themes as is more scalable and flexible. Distributed meta-scheduling algorithms have been studied for many years. Work in [43] proposed a wide-area scheduling system based on a local resource management system (LRMS) and a wide-area scheduler (WA). Each member of the site has to instantiate a) the LRMS which manage the local resources and b) the WA which achieves a global scheduling. Specifically, the WA scheduler contains two interfaces; firstly the scheduling manager to local schedulers and secondly, a grid scheduler to remote scheduling managers. The sharing of information based on a static file of addresses in which grid schedulers can access at any time. Similar to [43] authors in [37] presented a meta-scheduling mechanism called NWIRE (Net-Wide-Resources). The scheduler consists of a MetaManager who is responsible for controlling a set of domains and has access to the LRMS. The NWIRE considers several scheduling characteristics including existence of conventional schedulers, resource reservations and resource trades. In general NWIRE offers a high fault tolerance mechanism as the failure of a single trader will not affect the whole procedure.

In [1] a decentralised dynamic algorithm namely estimated load information scheduling algorithm is presented. 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 [1]. The aim is to increase the possibilities of gain load-balancing by estimation based on updated information after large time intervals. The ELISA basic concept is that at periodic intervals, called status exchange interval, the processors exchange their queue length and an estimate job arrival rate. The results presented in [1] have

shown that ELISA is an efficient solution for achieving load balancing in large distributed systems. The necessity for coordinated resource management in distributed systems is presented in [19]. The work presents a model namely federation of distributed resource traders and parallelise jobs submissions to user defined services. By coupling several resources or providers the resource trader acts similar to a meta-scheduler as the intermediate among consumers and providers. Several traders cooperate with each other in order to develop a federation of traders in which local users, clients and resources managed by each trader will trade resources. The results presented in [19] indicate that when using trader federation an improvement in the resolution times can achieved. However, this method doesn't present how data consistency is managed [26], as well as there is no discussion about the actual simulation environment.

Work presented in [41] demonstrates a distributed computing scheduling model which "adapts to changes in global resource usage" [41]. The key idea of the proposed meta-scheduler is to redundantly distribute each job to multiple sites, instead of sending the job to the most lightly loaded. Specifically, when a job is placed in multiple sites the possibility of effective backfilling is higher. The technique measures the average job turnaround time and average job slowdown. In [8] the authors present a model for connecting various Condor [22] work pools which yields to a self-organising flock of Condors. However, the model uses the Condor resource manager to schedule jobs to various idle resources, and invokes the flocking mechanism only in the case in which the machines are busy. Specifically, the scheme compares queue lengths, average pool utilisation and resource availability and creates a list of pools. The results show that the flocks can reduce the maximum job waiting time in the queue. Work in [2] presents a scheduling infrastructure based on the bag-of-tasks applications and called OurGrid. The OurGrid is a collection of peers constituting a community. Specifically, the system contains the following components; the Swan which is the software system for making possible access to resources from community members, the OGBroker which is the resource consumer brokering system and the OGPeer which is the mean to connect OGBroker to OurGrid and scheduling happens by the site's reputation and resource availability. In [31] authors discuss also a market-based resource allocation system in which based scheduling in auctions. Specifically, each resource provider or owner runs an auction for his resources. The meta-schedulers communicate with a Service Location Service (SLS) which contains an index of resource auctioneers. In SLS auctioneers record their status every thirty seconds. The bid for resources happened by the meta-schedulers who acts on behalf of their resources. However, with this solution resources can be under-utilised as meta-schedulers may bid always for a specific set of resources. This concludes to a coordination lacking of the meta-scheduling method. Work presented in [38] suggests two scheduling algorithms namely the modified ELISA (MELISA) [1] 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. Specifically, MELISA calculates the neighbouring nodes load by considering jobs arrival rates, service rate and node loads. However, in contrast with ELISA the jobs are transferring decision is based on the comparison of nodes load and not the queue length. To improve MELISA performance, the authors conclude that the load balancing on arrival method will balance the high job arrival rates.

The delegated matchmaking (DMM) approach presented in [27], is a novel delegated technique which allows the interconnection of several grids without requiring the operation of central control point. This happened by temporarily bind local resources to remote resources. Specifically, in this decentralised approach when a user cannot be satisfied at the local level, then through a delegated matchmaking procedure remote resources are added to the user transparently. The DMM utilise a hierarchical architecture in which resources in the same level may cooperate with each other. So, by delegating resources and not jobs the DMM aims to minimise the overhead caused by the management of jobs. The results of the simulation show that DMM can have significant performance and administrative advantages. However, this work raises questions of heterogeneity issues in large scale distributed settings. Also job failures and unmovable loads at the cluster level are not considered. The work in [13] presents 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. So they suggest that there is a need for new settings that will allow grid to evolve from local to global scale. Specifically, InterGrid suggests interlinking grid islands using peering arrangements. Thus, by providing a flexible and scalable construction a sustainable connection can happen among grids. In a similar vein the work in [15] evaluates the performance analysis of the InterGrid architecture by using conservative backfilling, multiple resource partition, least loaded resource policy and earliest start time policy. Finally, the results show that the average response time has been improved in the aforementioned evaluated scheduling algorithms. However authors in [32] suggest that this approach reflects a more economical view when business application support is the primary goal.

They present a decentralised model for addressing scheduling issues in federated grids. This solution propose 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. Authors suggest four algorithms that could be executed in the GridWay meta-scheduler namely; the static objective (SO), the dynamic objective (DO), the static objective and advance scheduling (SO-AS) and the dynamic objective and advance scheduling (DO-AS). Starting with the SO algorithm which aims to a higher throughput, an objective decides the number of jobs to be submitted to a host. The DO is a more complicated approach which determines objectives which are actually processed during the execution time. Finally, both SO-AS

and DO-AS share similar functionalities to the aforementioned SO and DO, however with one major difference. Specifically, jobs are advanced scheduled in desired resources without waiting for free nodes. Experimental results presented in [32] reveal that DO-AS is the best strategy as outperforms other solutions SO-AS in minimising makespan times. The last one is completely transparent to the users by not request for information. Its great advantage is that the method considers past performance requirements forecasts new objectives. Thus, authors suggest that this flexible method of DO-AS is fast enough to be used in realistic scheduling.

Work in [18] presents an Evolutionary Fuzzy System approach for identifying situation adaptive and robust algorithms for workload distribution in decentralised grids. Authors suggest a decoupled grid resource management system (GRMS) which decides the delegation of jobs from site to site. Jobs are submitted to the local resource management system (LRMS) as usually, however a submission component intercepts those and forward them to a local GRMS for further investigation. Authors in [18] further discuss that "the decision mechanism is established by using a Fuzzy controller system with flexible rule sets that are optimised using evolutionary computation, using a pair-wise training approach and performance metric-based rule base selection". This happens because in some cased resource utilisation, throughput and average response times remain confidential. This happens because of resources competitions or security reasons. Therefore, such information is not sharable during the scheduling process. The evaluated results are based on real world data and show that it is possible to exchange policies which lead to response time and utilisation improvements. Finally, the authors suggest that enhancement of performance can come from a stable basis for workload distribution.

In [36], authors have addressed the problem of broker selection in multiple grid scenarios by describing and evaluated several scheduling techniques. In particular, a system entity e.g. hosts and virtual organisations are represented as meta-brokers which might behave as gateways. Every scheduling method discussed in this work consists of the "bestBrokenRank" broker selection policy along with two different variants namely bestBrokerRank_AGGR (AGGR_SIMP and AGGR_CAT) policy and bestBrokerRank_SLOW policy. The first one utilises the resource information in aggregated forms as input, and the second one utilises the dynamic performance metric "broker average bounded slowdown". However, although the interoperable grid scenarios can improve workload executions and resource utilisation, this work did not address issues in matching time with aggregated resource information.

Authors in [8] discuss the problem of overloading by suggesting an alternative mean of resource selection called bidding. Since, there is no global information available in a dynamic environment (e.g. grid and cloud), bidding cannot facilitate optimum decision. For this reason, a resource selection heuristic method has been proposed in order to minimise the turnaround time in a non-reserved bidding based grid environment. The first heuristic is called random selection and the probability of selection is given by a mathematical formulae. The second is the minimum execution time-deterministic and selects resource providers with the minimum execution time. The third is called minimum execution time-probabilistic and the selection of a provider is proportional to the CPU capability. The fourth is the minimum completion time-deterministic is similar to the second heuristic with an added characteristic, selection happens according to the waiting time plus the execution time. Finally the fifth is the dissolve-probabilistic and selection of providers is inspired by the way ice cubes dissolve by calculating the proportion of the served workload to the whole workload. By conducted a series of experiments the authors claim that dissolve-probabilistic performs better than the other heuristics. However, this work didn't consider important scheduling issues which might affect performance, such as job workload, CPU capability, job execution deadlines, bandwidth and network features and dynamic availability of resources.

Authors in [47] introduce 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. In particular, the first called job submission phase finds the proper node from the scope of the overall grid (job distribution) and the dynamic scheduling phase, which aims to iteratively improving scheduling decisions. Its great difference when comparing with aforementioned approached is that aims to an overall performance improvement, rather than individual hosts performance boosting. The authors by conducting a series of experiments have shown significant results. First of all, by applying the CASA in a decentralised scheduling theme it could lead to the same amount of executed jobs comparing with the centralised solution. Also, job slowdown and waiting times have been dramatically improved. In addition, the authors claim that improvements were also noticed on the scheduling performance including response and waiting time and the messages overhead.

The CASA, in contrast with aforementioned algorithms, is based on contacted nodes' real time responses. However, the authors suggest that further enhancements should be considered to include backfilling methods and shortest job first. Also further experiments could be considered by using different grid workload traces in order to get a better understanding of the improved performance. In [35] authors present a scheduling strategy based on backfilling called JR-backfilling and resource selection policy called the SLOW-coordinated policy. The method uses dynamic performance information instead of job requirements and resource characteristics. The overall algorithm aims to the minimisation of the workload execution time, job waiting time, job response time, average bounded slowdown and to maximise the resource utilisation. Obtained results show that the JR-backfilling outperforms the FCFS and, in addition, SLOW-coordination performs better than the traditional matchmaking approaches in terms of workload execution

time etc. However, the FCFS approach is simple comparing with dynamic solution and more results are expected in order to compare the feasibility of the work.

Work discussed in [44] presents a job scheduling algorithm which considers the commercialisation and virtualisation characteristics of cloud computing based on the Berger Model. The model suggests distribution justice based on expectation states which study actors and evaluate their behaviour. To conclude, various scheduling categories applicable for a wide area of systems have been discussed. With regard to this work the aim is on the interoperable infrastructures and in particular scheduling user defined tasks in such high dynamic and distributed infrastructures. This is the reason why the attention has been focused on the meta-scheduling scheme. It has been found that the main part of the approaches constantly motivating from either a flexible and/or scalable and/or heterogeneous and/or dynamically changed infrastructure. Since interoperable infrastructures are a collection of sub environments, sudden variations can happen during scheduling, thus making essential that the aforementioned motivation to facilitate the form in which the inter-cloud should be considered [4], [5]. In this case, load coordination must happen automatically and distribution of user requests (either in the form of job tasks or services) must change in response to changes in the load. However, as the complexity and dynamics of the systems increased e.g. grid and clouds, the need for vigorously changed scheduling decisions have led developers to dynamic decentralised meta-scheduling methods. In such case, interoperable environment e.g. InterCloud could be composed by a pool of sub clouds which can join or leave the infrastructure at any time, thus behave dynamically. This clearly drives us to study scheduling solutions for dynamic environments in depth.

III. Discussion on Remarks

The meta-scheduling theme has proven to be a very promising approach because of its capability to handle efficiently scalability and flexibility issues in large scale resource pools. As presented in previous section the aim is on classifying relevant approached for identifying crucial characteristics that are relevant to the desired scenario, meta-scheduling in inter-infrastructures. It should be mentioned that each approach has been developed to address different requirements, thus meta-scheduling themes are classified according to their effectiveness in bridging the gap of resource amongst large scale and various size settings. Thus, it could be said that 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.

However, the scheduling results of both centralised and hierarchical topologies are important as they are always powerful in performance and simple in job requirements. From the early ages, e.g. 1998, centralised meta-scheduling has been studied in the simplest of the forms as the mean to schedule job tasks in a subset of processors in interconnected networks. Various efforts have made since then for achieving different requirements posed by different scenarios. The characteristics of centralised and hierarchical approaches are summarised in the list below and could lead to the identification of relevant concerns to various future scenario. As the aim of this study is the review of various scheduling approaches in the next section we present our classification study based on information extracted from the literature.

A. Centralised and Hierarchical scheduling remarks

Homogeneous pool of resources is usually assumed or heterogeneous pools as a crucial issue is frequently neglected in various cases e.g. [11], [6], [40], and [21]. However, several works include heterogeneity to their initial requirements e.g. [20], [23], [34]. Those solutions have been tested and benchmark results extracted from experiments show notable performance. On the other hand, this happens only in small scale environments with static scheduling objectives and no system dynamics consideration.

Interoperability among local schedulers is not usually considered in several works e.g. [11], [20], [6], [7], yet still in [55], [69] authors aim to an interoperable environments.

Dynamic-ness of the environment is the major issue neglected by numerous works e.g. [7], [6]. [11], [41] as all demonstrate results in static and not realistic scenarios. However, work in [35] presents a solution in which meta-queue jobs are pulled dynamically.

Geographical distribution between different pool of resources is not considered in most of the works e.g. [11], [6], [45], [41], [20] as all suggest that their experiments shows significant results in localised environment. However, the work in [23] considers geographically dissemination of resource, yet still the global dispatcher shows decreased performance in high workloads.

Inter-collaboration for job sharing among different infrastructures e.g. grid virtual organisations is usually ignored as the small scale-ness of the supposed environment doesn't allow such assumptions. Specifically works of [11], [6], [24], [23] do not aim of addressing the issue. However, work in [45] considers a multi-cluster environment and work in [29] extends the resource sharing to a setting composed from various grid virtual organisations.

Load-balancing is also an issue which is found to be not considered by most of the centralised works e.g. [41], [11], and [20]. However, the works in [34] and [23] have performed experiments and present their results from the scope of an overall load balancing mechanism.

Resource allocation mechanisms in centralised solutions have found to be driven by different scenarios. The most common of them aim to achieve a general fairness e.g. the market based scheduler as presented in [3]. Conversely, self-centred based driven solutions as in [23] could improve the performance of individual nodes, though such works don't aim to develop a wide and global resource provisioning.

Rescheduling concept and **Advance reservation mechanism** have been considered by the least of works e.g.

[45] utilises the backfilling scheduling algorithm and improved performance have been demonstrated. Similarly, the work [41] presents an advance reservation strategy that assigns meta-jobs data to specific local scheduler, thus jobs are matched with exclusively used resources.

Previous work delegations in the form of historical data are not considered by the majority of the works, although it could contain a future value for enhancing the rescheduling and advance reservation process. However, work in [41] tries to achieve a similar strategy by assigning jobs to specific resources based on a statistical consideration.

Security issues are usually ignored and resource managers are assumed to do the specific work. Usually, this issue is out of the scope of the meta-scheduling theme; however it cannot be neglected as it is one of the important parts of a comprehensive architectural model. The security problem gets worst when the system extends to a wider area resource pool in which a variety of attacks that be mounted against individual nodes could happen. To this extend, work in [21] offers a secure distributed resource management framework based on resource tickets, agents and resource managers that is effective in homogeneous systems.

To conclude the above characteristics are derived from the cross-correlation of various centralised and hierarchical scheduling approaches. However, those approaches have proven to be more appropriate solutions for small scale environments e.g. [11], [20], [6] etc. Thus, the decentralised/ distributed scheme aims of addressing relevant to previous and more complex requirements. Most of the decentralised meta-scheduling approaches include crucial characteristics towards wider-scheduling decisions in inter-collaborated environment as presented below.

B. Decentralised Scheduling Remarks

Heterogeneous pool of resources is recognised as one of the crucial subjects in various cases e.g. [35] [36], [58], [2], [19] and [32]. However, the literature study shows that tentative results from the aforementioned works confirm a low appreciation of the heterogeneity issue during experimentation. Homogeneous pool of resources, on the other hand, in the most of the approaches is not the specific scenario case. However, there are still works that either don't include heterogeneity or assume homogeneity to their requirements scenarios e.g. [43], [37], [47].

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 [43], temporary binding amongst resources and jobs [27] or by supporting fault tolerance mechanisms as in [34].

Dynamic-ness of the environment is a critical property when developing an interoperable meta-scheduler. Various work try to solve meta-scheduling issues derived from the unpredictability of a dynamic changing environment as in [32] which considering s past performance requirements for forecasting new objectives. Similarly, authors in [47] present a meta-scheduling tactic that doesn't expose internal node information and based on nodes' real time responses. Equally, work in [35] uses dynamic performance information instead of job requirements and resource characteristics. In contrast with those solutions, non-dynamic approaches such as [23] and [36] assume a steady-state setting during simulation. In the last one, authors suggest a delegated matchmaking procedure in which resources are matches temporarily to remote resources.

Geographical distribution of different pool of resources is considered in most of the works as they all include meta-scheduling for grid environments. Specifically, [18], [35], [36], [46], [47] etc. 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, works in [13] and [15] aiming to an inter-grid of interlinking grid collaborated islands using peering arrangements. Work in [47] present a more advance meta-scheduling algorithm for job scheduling among distributed grid nodes. Similarly, works in [46], [36], etc. aim to an inter-collaborated theme.

Load-balancing of different settings has been identified in various works such as in [35], [36], [47], [12] etc. Specifically, the increasing load balancing probability improves the performance of the overall environment. For example, in [1] the algorithm estimates the queue length of neighbouring nodes and then performs a rescheduling process. Likewise, in [38] calculates the neighbouring nodes load by considering job arrival rate, service rates and node loads. In this case, jobs are transferred based on the comparison of nodes load and not queue length.

Resource allocation mechanisms in decentralised solutions have found to be driven by different scenarios. In [8] the method connects various Condor pools based on a self-organising flock of Condors. In [2] scheduling happened by site's reputation and resource availability. A market based resource allocation model is discussed in [31] in which auction list of resources is attained by meta-schedulers who acts on behalf of their resources. Past performance information in the form of historical data are utilized by [32] to achieve a resource allocation mechanism. To conclude, various mechanisms exist in literature always based on the requirements of the specific scenario. For example in [44] authors present a scheduling algorithm which considers the commercialization and virtualization characteristics of cloud computing based on the Berger Model, thus it is more an economic driven setting within a single cloud rather an inter-cooperative intensive mechanism. Due to job scheduling in clouds two constraints are established aiming to fairness.

Rescheduling concept and **Advance reservation mechanism** is commonly used in various cases for iteratively improve the performance of the scheduling process. Specifically, in [47] authors claim that during a rescheduling phase a notable improvement has been observed in the scheduling performance. Equally, [32] suggests 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 significantly, especially in the case in which a large scale job input arrives in the scheduler.

Previous work delegations in the form of historical data are not considered by the majority of the works, although it could contain a future value for enhancing the rescheduling and advance reservation process. However, the work of [32] tries to achieve a similar strategy in which the method considers past performance requirements and might forecast new objectives. However, this is only adoptable for specific information system as requires training mechanism for forecasting performance. Alike, in [47] the method considers past job delegation records during the rescheduling process.

Security issues, similarly to the decentralised and hierarchical meta-scheduling topology are usually ignored and resource managers are assumed to do the specific work. Typically, this issue is out of the scope of the meta-scheduling theme. In the decentralised meta-scheduling the security problem includes more issues like information exposition during meta-schedulers collaboration.

Neighbouring collaboration is mainly the development of various size cliques that share commonalities in job requirements, while at the same time could belong or not at the same administration domain. Examples are the Condor pools in [8], and the grid islands in [13] and [15]. Both solutions could offer sustainable connections among different communities, however, unfairness among resources could lead to starvation and the dynamics could affect the certain connections.

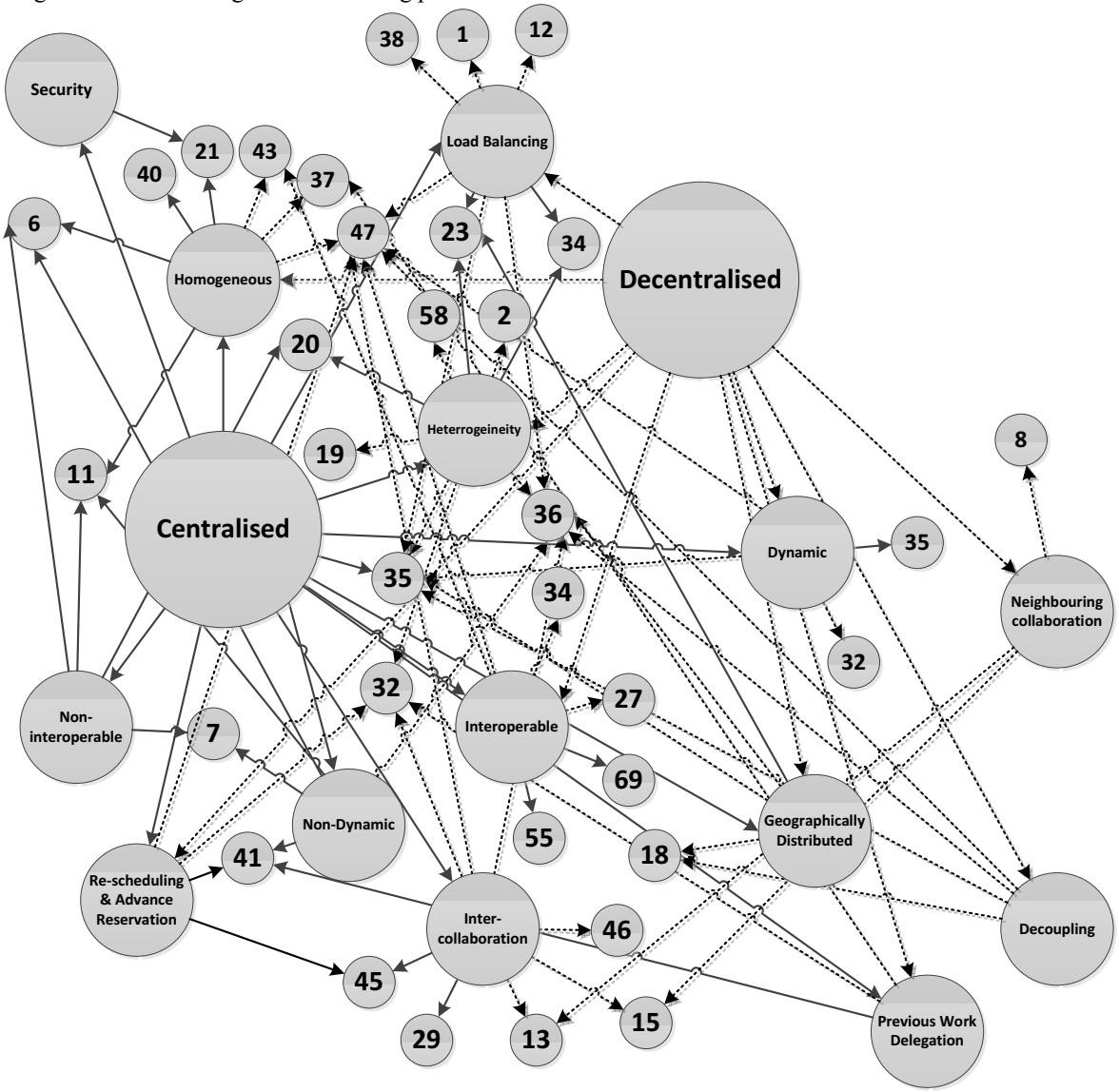

Figure 1: Mapping of reviewed approaches to their extracted characteristics
(key: numbers denote references from the referencing list)

Coupling of specific jobs to resources could lead to a temporarily improved performance setting as presented in [27] however as mentioned previously dynamics could affect the coupling relationships. A solution to this problem could be the advance reservation mechanisms for coupling jobs to resources as in [32], or local to remote resources as in [27] in a temporarily based for offering momentary boost of performance. **Decoupling**, on the other hand decides the delegation of jobs from site to site without connecting resources. Examples are the work in [18] in which jobs are submitted as normally from meta- to local scheduler, however a submission component redirects to a global resource manager for further inspection. Using an evolutionary computation method optimize workload exchanging. Similarly, [36] presents a policy that considers dynamic performance metrics, [35] based on backfilling uses dynamic performance information and finally [47] performs scheduling of jobs based on dynamic real time node responses. Those characteristics of decentralised meta-scheduling approaches are summarized in the list below and could lead to the identification of relevant concerns to each study's' specific scenario e.g. InterCloud. It should be mentioned, that the specific characteristics are derived from the cross-evaluation of various literature works.

Figure 1 demonstrates a mapping of various approaches, including centralised (including hierarchical) and decentralised, based on our discussion to their extracted characteristics. It shows topologies that are plotted to various characteristics which in turn are mapped to scheduling approaches. It should be mentioned that the numbers denote references from the referencing list. It is anticipated that the future meta-scheduling issues could be evaluated, assessed and integrated according to the review characteristics as derived from the cross-correlation mapping study.

IV. CONCLUSION

This work presents a state-of-the-art review of meta-scheduling related technologies and the analysis of their remarks. Specifically, a detailed evaluation of meta-scheduling literature approaches has been discussed including different topologies as in [4], [5] and [39]. The approaches presented herein recognise the most important characteristics that could be applicable to future interoperable designs. This is also so the contribution of this research study, by analysing various scheduling approached applicable to different topologies we conclude to the mapping of their characteristics. As the aim of this study was the review of various scheduling approaches the previous section presented our classification study based on information extracted from the literature. It is expected that future meta-scheduling approaches that aim to be applicable to several environments e.g. HPC, grids and clouds, could use this study as the basis for recognising their needs and eventually identify the most relevant scheduling approaches.

REFERENCES

[1] Anand, L., Ghose, D., and Mani V., ELISA: an estimated load information scheduling algorithm for distributed computing systems. Computers & Mathematics with Applications, 37(8):57-85, 1999.

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

[3] Auyoung, A., Chun, B., Snoeren, A., and Vahdat, A. (2004). Resource allocation in federated distributed computing infrastructures. In OASIS '04: 1st Workshop on Operating System and Architectural Support for the On-demand IT Infrastructure, Boston, MA, October, 2004.

[4] Bessis, N., Sotiriadis, S., Cristea, V., Pop, F., Towards inter-cloud schedulers: Modelling Requirements for Enabling Meta-Scheduling in Inter-Clouds and Inter-Enterprises, Third International Conference on Intelligent Networking and Collaborative Systems (INCOS 2011) , Nov 30 - Dec 2 2011, Fukuoka, Japan.

[5] Bessis, N., Sotiriadis, S., Cristea, V., and Pop, F., Towards inter-cloud schedulers: Modelling Requirements for Enabling Meta-Scheduling in Inter-Clouds and Inter-Enterprises, Third International Conference on Intelligent Networking and Collaborative Systems (INCOS 2011), Nov 30 - Dec 2 2011, Fukuoka, Japan .

[6] Bitten, C., Gehring, J., Schwiegelshohn, U., and Yahyapour, R., 2000. The NRW-Metacomputer-Building Blocks for A Worldwide Computational Grid. In Proceedings of the 9th Heterogeneous Computing Workshop (HCW '00). IEEE Computer Society, Washington, DC, USA, 31.

[7] Braun, T.D., Siegel, J.H., Beck, N., Lasislau Boloni, L., Maheswaran, M., Reuther, I.A., Robertson, P.J., Theys, D.M., Yao,B., Hensgen, D., and Freund, F.R., 2001. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *J. Parallel Distrib. Comput.* 61, 6 (June 2001), 810-837.

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

[9] 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), Volume: 6081/2010, Issue: LNCS 6081, Publisher: Springer, Pages: 13-31.

[10] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. and Brandic, I., Cloud computing and emerging it platforms: vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems 25 (6) (2009), pp. 599–616 10.1016/j.future.2008.12.001.

[11] Carroll, T.E. and Grosu, D., "An Incentive-Compatible Mechanism for Scheduling Non-Malleable Parallel Jobs with Individual Deadlines," Parallel Processing, 2008. ICPP '08. 37th International Conference on , vol., no., pp.107-114, 9-12 Sept. 2008.

[12] Christodoulopoulos, K., Sourlas, V., Mpakolas, I., and Varvarigos, E., A comparison of centralized and distributed meta-scheduling architectures for computation and communication tasks in Grid networks, Computer Communications, Volume 32, Issues 7-10, 28 May 2009, Pages 1172-1184, ISSN 0140-3664.

[13] De Assuncao, M., Buyya, R., and Venugopal, S., InterGrid: A case for internetworking islands of Grids, Concurrency and Computation: Practice and Experience 20 (8) (2008) 997–1024.

[14] De Assuncao, M., Costanzo, A., and Buyya, B., (2010). A cost-benefit analysis of using cloud computing to extend the capacity of clusters. Cluster Computing, Volume 13, Issue 3, September 2010, Pages 335 347 ISSN: 1386-7857.

[15] De Assuncao, M.D. and Buyya, R., Performance analysis of allocation policies for interGrid resource provisioning. Information and Software Technology, 51(1):42-55, 2009.

[16] Feigenbaum, J. and Shenker, S. (2002). Distributed algorithmic mechanism design: recent results and future directions. In DIALM '02: Proceedings of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications,

Atlanta, Georgia, USA, pages 1–13. ACM Press, New York, NY, USA.

[17] Feitelson, D.G., and Rudolph, L., 1995. Parallel Job Scheduling: Issues and Approaches. In Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing (IPPS '95), Dror G. Feitelson and Larry Rudolph (Eds.). Springer-Verlag, London, UK, 1-18. Approaches," in Job Scheduling Strategies for Parallel Processing (JSSPP), 1995, pp. 1–18, LNCS 949.

[18] Folling, A., Grimme, C., Lepping, J., and Papaspyrou, A., Decentralized grid scheduling with evolutionary fuzzy systems. In Job Scheduling Strategies for Parallel Processing, pages 16{36. Springer, 2009.

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

[20] Freund, R. F., Gherrity, M., Ambrosius, S., Campbell, M., Halderman, M., Hensgen, D., Keith, E., Kidd, T., Kussow, M., Lima, J. D., Mirabile, F., Moore, L., Rust, B., and Siegel, H. J. 1998. Scheduling Resources in Multi-User, Heterogeneous, Computing Environments with SmartNet. In Proceedings of the Seventh Heterogeneous Computing Workshop (HCW '98). IEEE Computer Society, Washington, DC, USA.

[21] Fu, Y., Chase, J., Chun, B., Schwab, S., and Vahdat, A. (2003). SHARP: an architecture for secure resource peering. In SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles, Bolton Landing, NY, USA, pages 133–148. ACM Press, New York, NY, USA.

[22] Globus website: http://www.globus.org/toolkit/, Accessed 10 November 2011.

[23] Grosu, D. and Chronopoulos, T. (2004). Algorithmic mechanism design for load balancing in distributed systems. In IEEE Transactions on Systems Man and Cybernetics Part B, pages 77–84. IEEE Computer Society, Los Alamitos, CA, USA.

[24] Guim F., and Corbalan, J., 2007. A job self-scheduling policy for HPC infrastructures. In Proceedings of the 13th international conference on Job scheduling strategies for parallel processing (JSSPP'07), Eitan Frachtenberg and Uwe Schwiegelshohn (Eds.). Springer-Verlag, Berlin, Heidelberg, 51-75.

[25] Hamscher, V., Schwiegelshohn, U., Streit, A., and Yahyapour, R., 2000. Evaluation of Job-Scheduling Strategies for Grid Computing. In Proceedings of the First IEEE/ACM International Workshop on Grid Computing (GRID '00), Rajkumar Buyya and Mark Baker (Eds.). Springer-Verlag, London, UK, 191-202.

[26] Ibarra, H. O., and Kim, E., C., 1977. Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors. J. ACM 24, 2 (April 1977), 280-289.

[27] Iosup, A. , Tannenbaum, T., Farrellee, M., Epema, D., and Livny, M., Inter-operating grids through delegated matchmaking. Scienti_c Programming, 16(2):233-253, 2008.

[28] Jackson, D., Snell, Q., and Clement, M., Core algorithms of the Maui scheduler. In Job Scheduling Strategies for Parallel Processing, pages 87-102. Springer, 2001.

[29] Kertesz, A., Farkas, Z., Kacsuk, P., and Kiss, T., Grid Interoperability by Multiple Broker Utilization and Meta-Broking. Grid Enabled Remote Instrumentation, pages 303-312, 2009.

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

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

[32] Leal, K., Huedo, E., and Llorente, I.M., A decentralized model for scheduling independent tasks in federated grids. Future Generation Computer Systems, 25(8):840-852, 2009. 21, 27.

[33] Mateescu, G., Gentzsch, W., and Ribbens, J.C., Hybrid Computing--Where HPC meets grid and Cloud Computing, Future Generation Computer Systems, Volume 27, Issue 5, May 2011, Pages 440-453, ISSN 0167-739X.

[34] Pinchak, C., Lu, P., and Goldenberg, M., 2002. Practical Heterogeneous Placeholder Scheduling in Overlay Metacomputers: Early Experiences. In Revised Papers from the 8th International Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP '02), Dror G. Feitelson, Larry Rudolph, and Uwe Schwiegelshohn (Eds.). Springer-Verlag, London, UK, UK, 205-228.

[35] Rodero, I., Guim, F., and Corbalan, J., 2009. Evaluation of Coordinated Grid Scheduling Strategies. In Proceedings of the 2009 11th IEEE International Conference on High Performance Computing and Communications (HPCC '09). IEEE Computer Society, Washington, DC, USA, 1-10.

[36] Rodero, I., Guim, F., Corbalan, J., Fong, L., and Sadjadi S. M., Grid broker selection strategies using aggregated resource information. Future Generation Computer Systems, 26(1):72-86, 2010. 21, 26.

[37] Schwiegelshohn, U., and Yahyapour, R., Resource allocation and scheduling in metasystems. In High-Performance Computing and Networking, pages 851-860. Springer, 1999.

[38] Shah, R., Veeravalli, B., and Misra, M., On the design of adaptive and decentralized load balancing algorithms with load estimation for computational grid environments. IEEE Transactions on parallel and distributed systems, 18(12):1675-1686, 2007.

[39] Sotiriadis, S., Bessis, N., and Antonopoulos, N., Towards inter-cloud schedulers: A survey of meta- scheduling approaches, Sixth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Oct 26-28 2011, Barcelona, Spain.

[40] Snell, Q., Clement, J. M., Jackson, D. B. and Gregory, C., 2000. The Performance Impact of Advance Reservation Meta-scheduling. In Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing (IPDPS '00/JSSPP '00), Dror G. Feitelson and Larry Rudolph (Eds.). Springer-Verlag, London, UK, 137-153.

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

[42] Teo, Y. M., Wang, X., and Gozali, J. P., 2004. A Compensation-based Scheduling Scheme for Grid Computing. In Proceedings of the High Performance Computing and Grid in Asia Pacific Region, Seventh International Conference (HPCASIA '04). IEEE Computer Society, Washington, DC, USA, 334-342.

[43] 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, pages 542-550, August 1996.

[44] Xu, B., Zhao, C., Hu, E., and Hu, B., Job scheduling algorithm based on Berger model in cloud environment, Advances in Engineering Software, Volume 42, Issue 7, July 2011, Pages 419-425, ISSN 0965-9978.

[45] Yue, J., Global backfilling scheduling in Multiclusters, Lecture Notes in Computer Science, Applied Computing, pages 232-239, vol. 3285 2004.

[46] Wang, C.M., Chen, H.M., Hsu, C.C., and Lee, J., Dynamic resource selection heuristics for a non-reserved bidding-based Grid environment. Future Generation Computer Systems, 26(2):183-197, 2010.

[47] Huang, Y., Bessis, N., Norrington, P., Kuonen, P., and Hirsbrunner, B., 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.