Towards inter-cloud schedulers:

A survey of meta-scheduling approaches

Stelios Sotiriadis, Nik Bessis, Nick Antonopoulos
School of Computing & Maths
University of Derby
Derby, United Kingdom
(s.sotiriadis, n.bessis, n.antonopoulos)@derby.ac.uk

***Abstract*— Cloud computing has emerged as one of the latest technologies for delivering on-demand advanced services over the internet. Various cloud providers have developed data-centres which are spread at several geographically locations, and are available for utilization from internet users. However, as the number of resource consumers is increasing significantly, it becomes apparent that the capacity-oriented clouds require coming together and agreeing on common acting behaviours for improving the quality of service, hence providing an overall optimal load allocation. In this direction, current solutions do not support a coordinated distribution of different cloud workloads. Even geographically distributed data-centres from the same vendor (e.g. Amazon) don't support a seamless mechanic for balancing hosted services as the users require indicating their selected hosts' location. To answer this limitation, a recently emerged inter-clouds notion comes to expand cloud capabilities and to offer an improved sharing paradigm of workloads. Herein we present a state-of-the-art review with a particular focus on the adoptability of current meta-schedulers for managing workloads, towards the inter-cloud era. Specifically, by exploiting inter-cloud requirements (e.g. flexibility, geographically distribution etc.) we evaluate various meta-schedulers for future inter-clouds.**

Keywords: Clouds, Inter-clouds, Scheduling algorithms, Meta-scheduling

I. INTRODUCTION

Cloud computing, in the latest years, has emerged as one of the most important technologies for delivering on-demand advanced services via the public internet [4]. A variety of cloud vendors e.g. Amazon develop their data-centres which could be spread at different geographically locations, and are available for utilisation from a worldwide set of internet users. However, as the number of resource consumers is increasing, it becomes apparent that the capacity-oriented clouds require coming together and agreeing on common acting behaviours for improving their quality of service, thus providing an optimization of their overall aggregated workloads. This prospect is underlined by [4] who suggests that current efforts do not support a coordinated distribution of different clouds workloads. To answer this limitation, a recently emerged notion called inter-clouds [11] comes to expand cloud capabilities by providing a flexible initiative for sharing resources. Specifically, inter-clouds form a pool of collaborated sub-clouds or sub-resources (e.g. grids) similar to a dynamic distributed system. Those systems always emerge new research questions towards flexible, heterogeneous and scalable scheduling decisions [5]. It has been suggested by [25] that scheduling has a major effect on systems' performance, and especially in workloads balancing in large systems, when a huge number of requested processes arrived at a scheduler, so the last one decides which to activate and when.

Typically, in the simplest form, a scheduler receives jobs, selects available resources according to availability and checks performance criteria to plan jobs to resources [25]. In an inter-cloud environment the complexity of scheduling is increased as the dynamics of the system are manifold. In this work, we aim to identify scheduling approaches of dynamic and scalable distributed systems to facilitate a mapping of current meta-schedulers to inter-clouds characteristics. This includes a state-of-the-art review of the distributed (meta-) heuristic (meta-) schedulers emerged in the latest years.

Section II presents the inter-cloud need and the general requirements for inter-clouds (section III) with the aim of addressing the importance of the scheduling issue. Then we discuss the local and meta-schedulers differences (section IV) and their topologies (section V). The rest of the paper contains the discussion of literature approaches (section VI), the evaluation of each approach in the basis of inter-cloud characteristics (section VII) and the conclusion (section VIII).

II. THE INTER-CLOUD NEED

Quite a lot of resource providers develop clouds with the aim of delivering high quality of service (QoS) in the context of work-load balancing, service response time and service cost. However, cloud resources (hardware and software) are usually limited in terms of serving capability. This problem gets worst when the ability of a single cloud provider to offer services to massive users' demands is bounded to the data-centre (DC) capacity. As clouds are usually DC facility oriented [8], a lack of providing high level of data redundancy could be observed [4]. Therefore, as the number of resource consumers increased the analogy of resource provision to consumption is getting non-balanced and the overall potential for improved QoS is restricted to the cloud boundaries. Thus, it is apparent that clouds could come together and lead to a form of an inter-cloud infrastructure in which sharing is motivated by an overall scope of performance boosting rather than nodes' self-motivated.

978-0-7695-4531-8/11 $26.00 © 2011 IEEE
DOI 10.1109/3PGCIC.2011.19

The growing interest in inter-cloud computing is highlighted by the biggest vendors in the area such as HP, Intel, Yahoo, etc. [4]. It is noticeable that their innovative efforts have led to the establishment of a federation of collaborated clouds with joint initiatives. This vendor effort of inter-clouds has a specific control plane rather that a setting based on future standards and open interfaces.

In general, inter-clouds necessity and adaptability could be highlighted by the area of disaster management in which a collaborated effort could lead to significant improved results. A simple example includes that clouds around the world can be utilized to backup and restore services from damaged data-centres. In reality, to support service provider cooperation, it is essential to establish communication among geographically distributed sites. Current efforts in this direction show that even the most famous vendors don't support scalable load balancing thus it observed a shredding of services among their data-centres [4]. Authors in [4] suggest that biggest providers, such as Amazon, request from their clients to select their best location for hosting services. This implies to a low QoS level as it is difficult for a user to decide its best cloud location. Consequently, in such system the performance is related to users' random decisions. So, standards for an inter-cloud environment includes that offered services from cloud providers must be a coordinated effort which dynamically scaled in order to offer a good quality of experiences to users [4].

For achieving aforesaid goals we present a study with regards to inter-clouds. This area can be considered as a wide research region in which issues such as resource discovery, allocation and scheduling are the most important. In this work we only focus on the job scheduling models for distributed environments (grids and clouds) so deliberately, we aim of identifying schedulers adoptable for inter-clouds. Before that, it is essential to analyze the requirements for inter-clouds in the scheduling area.

III. REQUIREMENTS FOR INTER-CLOUDS

The concept of connecting resources together (meta-computing) has been studied extensible, especially in the area of grid. For the majority of meta-computing systems scheduling is not a specific problem but a set of problem [5], [25] due to the different needs and the many characteristics of each group of resources. Similarly, scheduling among inter-clouds can be seen as a meta-computing scheduling mechanism in which a pool of inter-connected resources forming a community. Job scheduling algorithm in such environment is a complex decision mainly based on the nature of the inter-cloud to scale dynamically.

In large systems, resources behaviour is unpredictable as nodes can join or leave at any time. Besides that, the heterogeneity of resources add more complexity since resources can be ranging from laptops to supercomputers and so on. Thus, job scheduling in such system is also varied as jobs have different needs for their execution in terms of software and hardware. Furthermore, the large scale of the environment exposes a large number of local schedulers that require interacting with each other for optimizing the large number of submitted jobs. At last, resources could belong to several owners, so resource sharing based on service level agreements (SLAs) among vendors and between users and vendors. These agreements can change dynamically and could affect the availability of resources.

So, the dynamics within an inter-cloud system are principally determined by a) the dynamics of job execution, in which a job alters its status e.g. a highest priority job arrives, and b) the dynamics of resources in which a resource alters its primary status e.g. resource workload vary over time in unpredictable manner [4]. Those aspects characterize an environment as static or dynamic and it is it appears that inter-clouds have dynamics.

During the years scheduling within dynamic and uncertain distributed environments has been a hot research topic. A great variety of algorithms, theories and tools have been proposed by [1], [10], [12], [14] and others (section VI), aiming to a more flexible and efficient environment. However, when things come to reality it turns out to be impossible to design such tools [15]. The clear problem is that the actual requirements are not known in advance and depending on initial conditions and chosen parameters at real time processing. So the method for developing a flexible scheduler should be shelf-adaptive and fully automated with the aim of hiding complexity from the end-users. Also, an advanced solution to this direction could be the utilization of past nodes experiences from previous works in the form of historical data. A more detailed discussion of inter-clouds key characteristics is described afterwards (section VII).

The following section compares schedulers (local and meta-) for identifying inter-cloud scheduling applicability.

IV. LOCAL VS. META-SCHEDULING

Local and meta-schedulers (dynamic or not, centralized or decentralized) are the two fundamental solutions for scheduling in large scale distributed system e.g. grids and clouds [25]. Both aim to resource allocation and management, howbeit each one from a different perspective. The local scheduler is used at the cluster level scheduling, normally to achieve load balancing [5]. Meta-schedulers are used to assign user jobs to resources based upon user defined requirements [4]. In decentralized settings meta-schedulers put forward autonomously decisions which are transparent to the user by constituting the communication bus among different systems' local schedulers. Thus, a solution for high level and complex scheduling among inter-clouds can happen by bridging the gap of inter-communication among local grid or cloud resource managers using meta-schedulers.

During the years, meta-schedulers have been evolved in order to fill the gap of resource sharing within each local administrative domain. The scheduling in meta-computing was a challenging area for researchers mainly due to new additional requirements posed by the promising innovative technologies emerged in recent years (e.g. cloud, utility computing) [12].

One of the most important concerns is always the fact that different ownership of resources lead to different topologies. In the next section topologies are classified based on [20] and [25] into centralized scheduling, hierarchical structure, and decentralized scheduling.

V. META-SCHEDULING TOPOLOGIES

Scheduling topology is defined in the way in which resources are managed, organized, and administrated. In this context three meta-topologies are identified; the centralized, hierarchical and decentralized (distributed) as follows:

Centralized: Herein meta-scheduling happens directly by a central instance which maintains information of all resources [20], [25]. Each time new jobs are submitted; the centralized meta-scheduler sends the jobs for execution or arranges jobs in a queue.

Hierarchical: This meta-scheduling scheme is similar to the aforementioned centralized scheduling. In this setting jobs are submitted to a central instance of the scheduler and then jobs are directed to local queues [20].

Decentralized: This 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, compared to centralized and hierarchical themes as it is more scalable and flexible. Specifically, the meta-scheduler could have a real-time and instantaneous knowledge of the environment.

In centralized and hierarchical schemes the schedulers have a complete knowledge of the actual resource infrastructure which is unrealistic within an inter-cloud. This includes the number of hosts, number of jobs submitted, the workload of each hosts, and the topology of the system. Dissimilar, in the decentralized theme, this information is incomplete and the jobs received from the meta-scheduler are assigned to the local scheduler in 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 utilization, thus leading to a global load equilibrium as required in inter-clouds. So, the meta-scheduling scheme has better adaptability to dynamics and unpredictability of inter-clouds.

VI. DISCUSSION OF APPROACHES

In this section we present the functionality of eighteen existing meta-schedulers extracted from the literature which in the best to our knowledge are the most common used. A more detailed assessment (advantages, drawbacks and inter-cloud adaptability) of each approach is presented in section VII.

[24] proposes a wide-area scheduling system based on a local resource management systems (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. 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 anytime.

[20] presents a meta-scheduling mechanism called NWIRE (Net-Wide-Resources). The scheduler consists of a MetaManager who is responsible for controlling a set of domains (MetaDomains) and have access to the ResourceManager; the LRMS. The NWIRE considers several scheduling characteristics including existence of conventional schedulers and resource reservations and trade resources based on the economic mechanism.

[1] discusses a decentralized 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 aim is to increase the possibilities of gain load-balancing by estimation based on updated information after large time intervals. The ELISA [1] basic concept is that at specific periodic intervals the processors exchange their queue length and thus they estimate job arrival rate.

[10] presents a model namely federation of distributed resource traders and parallelize jobs submissions to user defined services. By coupling several resources to providers the resource trader acts similar to a meta-scheduler as the intermediary among consumers and providers. Thus, traders cooperate with each other in order to develop a federation of traders in which local users and resources managed by each trader will trade resources. Specifically, the collaboration of traders happens with the aim of maximizing a trading function. The trader contains two interfaces; the first one to the local scheduler and the second as a remote interface to other traders. Within this cooperative setting traders can negotiate for various parameters e.g. response time.

[22] demonstrates a distributed computing scheduling model which "adapts to changes in global resource usage" [21]. 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 (jobs to move to the front of a queue in order to fill any space created by a different algorithm) is higher. The actual algorithm requires minimal data and decides scheduling on current global picture of the system. This solution slightly decreases the overall performance.

[3] presents a model for connecting various Condor work pools which yields to a self-organizing flock of Condors. This work is more focused in the area of resource discovery by using a P2P routing Pastry model. 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. The results show that the flocks can reduce the maximum job waiting time in the queue.

[2] proposes a scheduling infrastructure based on the bag-of-tasks (BOT) applications called OurGrid. The OurGrid is a collection of peers constituting a community. Specifically, the system contains 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. Then scheduling happens by the site's reputation and availability.

[16] 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 meta-schedulers communicate with a Service Location Service (SLS) which contains an index of resource auctioneers. The bid for resources happened by the meta-schedulers who acts behalf of their resources. However, with this solution resources can be under-utilized as meta-schedulers may bid always for a specific set of resources. This concludes to a coordination lacking of the meta-scheduling.

[21] suggests two scheduling algorithms namely the modified ELISA (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. Specifically, MELISA calculates the neighbouring nodes load by considering jobs arrival rates, service rate and node loads. However, in contrast with ELISA [1] the job transferring is based on the comparison of nodes load and not the queue length. To improve MELISA performance, the authors concludes that the load balancing on arrival method will balance the high job arrival rates.

[14] 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. Specifically, in this decentralized approach when a user cannot be satisfied at the local level, then through a DMM procedure remote resources are added to the user transparently. The DMM utilizes a hierarchical architecture in which resources in the same level may cooperate with each other.

[7] 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. They suggest that there is a need for new settings that will allow grid to evolve from local to global scale. Specifically, InterGrid interlinks grid islands using peering arrangements by providing a flexible and scalable construction of a sustainable connection among grids. This happens by the use of InterGrid Gateways (IGGs) which allows a cross collaboration among islands. IGGs have decided arrangements among them and can perform resource allocation to different grids. [6] evaluates the performance and shows that the average response time has been improved in the aforementioned evaluated scheduling algorithms.

[17] presents a decentralized model for addressing scheduling issues in federated grids. This solution proposes the utilization of the GridWay; a meta-scheduler to each grid infrastructure of the federated grid. The method is an alternative to the centralized setting. The 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). Finally they suggest that the flexible method of DO-AS is fast enough to be used in realistic scheduling.

[9] discusses an Evolutionary Fuzzy System approach for identifying situation adaptive and robust algorithms for workload distribution in decentralized grids. The authors suggest a decoupled grid resource management system (GRMS) which decides the delegation of jobs from site to site. Jobs are submitted to the LRMS as usually, but a submission component intercepts those and forwards them to a local GRMS for further investigation. The evaluated results are based on real world data and show that it is possible to exchange policies which lead to response time and utilization improvements. The authors suggest that performance enhancement come from a stable basis of workload distribution.

[18] 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. Authors, claim that performance is not penalized significantly, and better results come by using the dynamic performance information. Although, the interoperable grid scenarios can improve workload executions and resource utilization, issues in matching time with aggregated resource information haven't discussed.

[23] suggests the problem of overloading by suggesting an alternative mean of resource selection called bidding. Authors claim that 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 approach has been proposed in order to minimize the turnaround time in a non-reserved bidding based grid environment. By conducted a series of experiments they claim that dissolve-probabilistic heuristics [23] performs better than the other selected heuristics. However, this work didn't consider important scheduling issues which might affect performance, such as job workload, CPU capability, job execution deadlines, network features and dynamic availability of resources.

[12] introduces a decentralized 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 decentralized distributed nodes. The first one, job submission phase, finds the proper node from the scope of the overall grid and the second one, the dynamic scheduling phase, aims to iteratively improving scheduling decisions. CASA great difference when comparing with the aforementioned approaches is that it 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 decentralized scheduling setting could lead to the same amount of executed jobs comparing with the centralized solution. Also, job slowdown and waiting times have been dramatically improved. This happens majorly because the model doesn't request detailed resource information from the resources. 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, they suggest that further enhancements should be considered to include backfilling methods and shortest job first.

[19] presents 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 minimization of the workload execution time, job waiting time, job response time, average bounded slowdown and to maximize the resource utilization. Obtained results show that the JR-backfilling outperforms the first come first served (FCFS) and, in addition, SLOW-coordination performs better than the traditional matchmaking approaches in terms of workload execution time. Yet, the FCFS approach is simple comparing with more dynamic solution. Also, authors suggest that the method performs better in a homogeneous environment rather than a heterogeneous setting.

[13] introduces a job scheduling algorithm which considers the commercialization and virtualization characteristics of cloud computing based on the Berger Model [13]. The model suggests distribution justice based on expectation states which study actors and evaluate their behaviour. Authors suggest that "the basic idea of distributive justice is that individual in social system can judge its own gained resources to be fair or not through distribution relations comparison between itself and other ordinary person in referential structure" [13]. Due to job scheduling in clouds two constraints are established aiming to fairness. The authors have validated their method in a simulation test-bed and results show better fairness. But, system dynamics have not been fully considered as it is parallelized to a large cluster base setting and not to an inter-cloud. The heterogeneity of cloud data-centres is assumed that it is hidden from the virtual machines.

In the next section, we evaluate and classify meta-scheduling according to their adaptability to inter-clouds. We aim to the most relevant features that current meta-schedulers target to achieve and inter-clouds require having.

VII. Evaluation of Reviewed Approaches

In the previous section we have discussed various categories for scheduling for a wide area of systems. With regards to this work, we aim on the inter-cloud environment and in particular scheduling tasks in high dynamic and distributed infrastructures. For that reason our attention has been focused on the meta-scheduling scheme as the most appropriate solution for flexible and decentralized scheduling. Thus, during this study, it has been found that the main part of the meta-approaches [12], [7], [25] constantly motivating from a flexible and/or scalable and/or heterogeneous and/or dynamically changed infrastructure. Since inter-clouds is a collection of sub clouds, sudden variations can happen during scheduling, thus making essential that the aforementioned motivation facilitates the form in which the inter-clouds should be considered. In this case, the workload coordination must occur automatically and distribution of user requests (either in the form of job tasks or services) must change in response to changes of the workload [4].

Whatever the case may be, static or dynamic, schedulers for distributed systems are important part of inter-clouds scheduling and are classified to local and meta-schedulers with regards to their operational environment. In the simplest of the forms, scheduling in a cluster base system uses local schedulers to control the resource allocation and management. Usually, the aim of local schedulers is to achieve a better load balancing. Meta-schedulers, on the other hand, are normally used for performing scheduling in grids. They are placed on the top of local schedulers and are used to assign jobs to resources based on a great variety of criteria. Through a collaboration of local and meta-schedulers a better scheduling decision can be observed [12] aimed to a dynamic and heterogeneous setting.

The following sections present an elaboration of the existing works extracted from the literature and a discussion of the following:

a) The essential characteristics of literature approaches for distributed systems e.g. flexibility, scalability, interoperability, heterogeneity etc (table I)

b) The essential characteristics of inter-clouds and the adaptability of the distributed system to the key inter-clouds requirements (table II) based on the drawbacks of approaches for inter-clouds on the basis of existing approaches.

c) The adaptable approaches which rise expectation towards inter-cloud meta-scheduling.

A. Charactesistics of Meta-scheduling Approaches in Distributed Systems

The general characteristics of distributed systems include flexibility, scalability, interoperability, heterogeneity, local autonomy, load balancing, information exposing, use of real-time data and scheduling considering history records [5], [25]. It should be mentioned that those characteristics are relevant to the aim of each meta-scheduler and its operational environment. For instance, in approach [2] heterogeneity and local autonomy is not considered as an important factor for this scenario. In fact, this is not a drawback as their system behaves well with minor problems, but we may consider it as an important drawback for inter-clouds.

By extracting current approaches characteristics we don't aim to their general evaluation but we intend to correlate them with the inter-clouds scheduling. Eventually, by this correlation we will describe which characteristics are analogous to the inter-cloud key requirements extracted from [4], [11], and [12]. Thus, which approaches can be utilized in future inter-cloud meta-scheduling. Table I (ND: not discussed, NA: not applicable) presents a correlation of these characteristics to each meta-scheduling approach of the literature. The columns describe the dynamic characteristics of distributed systems e.g. flexibility etc. which are the main target of each approach.

TABLE I. CHARACTERISTICS OF META-SCHEDULING APPROACHES IN DISTRIBUTED SYSTEMS

No	Characteristics / Meta-scheduler	Approach	Flexibility	Scalability	Interoper-ability	Hetero-geinity	Local autonomy	Load balancing	Information exposing	Real-time data	Scheduling history records
1	Wide-Area federated scheduler	[24]	ND	Yes	ND	ND	Yes	ND	NA	NA	No
2	Meta-domain scheduler	[20]	Yes	ND	ND	ND	ND	ND	NA	NA	NA
3	Distributed system scheduler	[1]	Yes	Yes	ND	Low	ND	Yes	Yes	NA	NA
4	Trader federation	[10]	ND	Yes	ND	Low	ND	ND	ND	NA	NA
5	Computational grids	[12]	ND	ND	ND	ND	Yes	Yes	No	NA	NA
6	Condor flocks	[3]	Yes	Yes	ND	ND	No	Low	Yes	NA	NA
7	Peers bokering	[2]	ND	Yes	Yes	Yes	No	Yes	Yes	NA	Yes
8	Auctioneer	[16]	Nd	Yes	ND	ND	ND	ND	ND	NA	ND
9	Distributed system scheduler	[21]	Yes	Yes	ND	Low	ND	Yes	Yes	NA	ND
10	Inter-operating grids	[14]	No	Yes	Yes	No	ND	Yes	ND	NA	ND
11	InterGrid performance analysis	[6], [7]	Yes	Yes	Yes	No	ND	ND	ND	NA	ND
12	Grid federation	[17]	ND	Yes	Yes	Low	Yes	ND	No	NA	Yes
13	Grid resource management system	[9]	ND	ND	Yes	Low	Yes	Yes	ND	NA	ND
14	Meta-broker model	[19]	ND	ND	ND	Low	ND	Yes	ND	NA	ND
15	Meta-broker model	[18]	ND	ND	Yes	Low	ND	Yes	ND	NA	ND
16	Bidding-base grid	[23]	ND	Yes	ND	Low	ND	ND	ND	NA	ND
17	Grid meta-scheduler	[12]	Yes	Yes	ND	ND	Yes	Yes	No	Yes	Yes
18	Cloud Berger scheduler	[13]	ND	ND	No	Yes	ND	ND	ND	NA	ND
		ND: not discussed		NA: not applied							

B. *Requirements for Inter-clouds*

Inter-clouds and distributed systems share various characteristics in the area of meta-scheduling. However, the key requirement is to achieve an optimized scheduling performance of the overall inter-cloud instead of individual participation nodes optimality. Thus the most important characteristics [4], [8], [12] are:

a) The management of unpredictability (dynamics)
b) The heterogeinity of resources
c) The geographically distribution of resources
d) The variation of job requirements
e) The compatibility on different SLAs
f) The rescheduling support

Table II illustrates those characteristics and the benchmarks used for optimizing each approach. Through this table we aim of addressing inter-cloud to each technique's characteristic. Then throughout this correlation an analogy of approach characteristic to inter-clouds drawback will be presented. So, we present a brief evaluation of the most important drawbacks of each literature approach from the scope of an inter-cloud environment as follows.

In [24] a theoretical model is presented without job history and job preferences support with low unpredictability. In [20] the rescheduling is not considered, while unpredictability and heterogeneity is low. In [1] information exchanging is hard to determine as also privacies are exposed and variation of job requirements are not discussed. In [10] dynamics are not considered adequately. [22] is a lightly loaded sites solution in which performance decreases in heavy workloads. In [3] the comparison of queue lengths exposes privacy and no heterogeinity considered. [2] is a promising solution wich manage dynamics however as scale grows job become less prioritized, thus could lead to starvation. In [16] there is a lack of coordination and heterogeinity in which resources bids' could aim to over-utilize resources. [21] is adaptable to dynamics, however, with low heterogeinity.

In [14] dynamics and heterogeneity are ignored and a steady state and identical processors is assumed. In [6] dynamics could affect "grid islands" connections and brokers are self and not global interested. [17] rises expectations as scheduling is based on dynamic and history records, however, is adoptable on specific information systems as well as overhead could observed during training. In [9] low dynamics and information sharing during competition have been observed. In [19], the method performs better in homogeneous settings. In [18] the work aims to load balancing of different grids, improve workloads and resource utilization, but fundamentally, a stable infrastructure was assumed. In [23], scheduling decision can change over time, although low unpredictability management and heterogeinity has been observed. Solutions [12] and [13] are described in detail in next section.

In table II we have discussed a brief evaluation of the most relevant characteristics of each approach in parallel to the inter-cloud setting. It is anticipated that for different environments those approaches could behave better or worse, but a general evaluation is not the aim of this study.

C. *Requirements of Approaches for Inter-clouds and their Classification in low, medium and high candidacy*

Herein we present a discussion of the adaptability level of meta-scheduling approaches for inter-clouds. For this reason we classify them to low, medium and high level candidate solutions based on the approach number of table II (first column number). Obviously, medium and high candidates' solutions contain the vital requirements that match with the inter-clouds.

Starting as low candidates we classify in this category approaches 1-6, 8, 10,11,13,14 and 16. The major reason for the most of them is the low management of dynamics. In addition heterogeneity is not fully considered and most of these approaches aim to a fundamental general stable environment without variation on job requirements. In addition history data have not been considered at all, and all solution based on user-defined data.

As medium candidate solutions we characterize 7, 9, 12 and 15, each one for a different reason. Approach 7, is a total decentralized approach in which nodes (peers) keep track of local balance for each known peer based on past interactions. Furthermore, the peers perform scheduling through a peer brokering to achieve heterogeneity and scalability. However, the variation of job requirements is low as well rescheduling is not considered at all. The evaluation metrics used for this approach is the number of resources gained by the brokers based on depts. As the dept grows, on the other hand, the jobs become more prioritized, thus this could lead to starvation.

Approach 9, calculates the neighbouring nodes load based on the comparison of the local workload. This method performs better in large scale systems and aims to an improved global equilibrium. However, adaptability to dynamics cannot be guaranteed as well as heterogeneity is not fully considered. In 12, a meta-scheduler aims to a federation of grids with minimum make-span and no job requirements. This is an important advantage as there is no processing speed, jobs lengths and information exposing of remote nodes. In addition the method considers past performance requirements. However, this is a system specific infrastructure of federated and not inter-collaborated resources so heterogeneity has not been considered. Also, when the environment extends to a bigger scale, overhead during training is increased. In 15, a meta-broker selection process is presented in multiple grid interoperable scenarios. The work aims to load balancing of different grids and improve workloads and resource utilization. However, since the infrastructure is assumed as stable results are unrealistic for inter-clouds.

To conclude medium candidate solutions match some of their requirements to the required ones from the inter-cloud scope, yet, important architectural issues (dynamics and heterogeneity not fundamentally addressed) possibly will make them problematic for future developments.

As high candidates solutions could be the 17 and 18, both aim to cloud meta-scheduling. The first one (17) could offer significant advantages to the inter-cloud scheduling concept as follows:

a) The CASA applied in a decentralized theme could lead to the same amount of executed jobs comparing with the centralized solution.

b) It is based on nodes' real time responses (dynamic).

c) The approach doesn't request and/or expose information from resources.

d) Simulation shows that job slowdown, waiting times, response times and messages overhead values have been significantly improved.

e) Encouraging results serve as the motivation for applying CASA to cloud computing.

f) Finally, CASA aims to an overall performance improvement, rather that individual hosts performance boosting.

However, improvements could be considered as follows:

g) The BestFit is considered as a simple conventional scheme for presenting above metrics, thus a better evaluation could lead to a better understanding of the model.

h) Approaches such as backfilling and shortest job first could improve performance.

i) Heterogeneity is not fully discussed thus should/could be addressed when utilizing virtualization technology.

j) Rescheduling has not been fully considered as the mean to improve meta-scheduling performance.

Work 18, presents a job scheduling algorithm which considers the commercialization and virtualization requirements of cloud computing based on the Berger Model [13]. Due to job scheduling in clouds two constraints are established aiming to fairness within a cloud. A significant advantage to the inter-cloud scheduling concept includes:

a) The method has been evaluated in a simulation test-bed and results show better fairness thus this model could be used in either competitive or cooperative clouds.

b) The authors discuss that scheduling based on cloud VMs could lead to a high heterogeneous environment as complexity is hidden from the virtual machines

TABLE II. REQUIREMENTS FOR META-SCHEDULING IN INTER-CLOUDS AND MAPPING OF TABLE I APPROACHES TO INTER-CLOUD

No	*Infrastructure* / ***Requirements***	Unpredictabilty managent	Heterogeneity	Geographically distribution	Variation of job requirements	SLAs compatibility	Rescheduling	Benchmarks	Inter-cloud adaptability
1	Wide-Area federated scheduler [24]	Low	ND	No	ND	NA	NA	Theoretical model	Low
2	Meta-domain scheduler [20]	Low	ND	ND	ND	NA	Low	Weighted response times	Low
3	Distributed system scheduler [1]	Low	Low	ND	ND	NA	Yes	Response times	Low
4	Trader federation 10	Low	Low	ND	ND	NA	NA	Response times	Low
5	Computational grids [12]	Low	ND	Yes	ND	NA	NA	Turnaround, slowdown times	Low
6	Condor flocks [3]	Low	ND	Yes	ND	NA	NA	Waiting times	Low
7	**Peers bokering [2]**	**Medium**	**Yes**	**Yes**	**Low**	**ND**	**NA**	**Resource gain**	**Medium**
8	Auctioneer [16]	Low	ND	Yes	Low	ND	NA	Job throughput	Low
9	**Distributed system scheduler [21]**	**Medium**	**Low**	**ND**	**ND**	**NA**	**NA**	**Response times, makespan**	**Medium**
10	Inter-operating grids [14]	Low	Low	Yes	ND	NA	NA	Waiting times, slowdown	Low
11	InterGrid performance analysis [6], [7]	Low	Low	Yes	ND	NA	NA	Response times	Low
12	**Grid federation [17]**	**Low**	**Low**	**Yes**	**ND**	**NA**	**NA**	**Makespan times**	**Medium**
13	Grid resource management system [9]	Low	Low	Yes	ND	NA	NA	Weighted response times	Low
14	Meta-broker model [19]	Low	Low	Yes	ND	ND	NA	Workload, waiting, response times	Low
15	**Meta-broker model [18]**	**Low**	**Low**	**Yes**	**ND**	**ND**	**NA**	**Waiting time, bounded slowdown**	**Medium**
16	Bidding-base grid [23]	Low	Low	Low	ND	ND	NA	Turnaround, response times	Low
17	**Grid meta-scheduler [12]**	**Medium**	**Low**	**Yes**	**Yes**	**ND**	**NA**	**Job slowdown, waiting times**	**High**
18	**Cloud Berger scheduler [13]**	**Low**	**Low**	**ND**	**ND**	**ND**	**NA**	**Completion, execution times**	**High**
		ND: not discussed	NA: not applied						

However, this approach has some disadvantages as it is more an economic approach within a single rather a global view of clouds. Also the method doesn't aim to an interoperable setting of clouds as well as system dynamics haven't been considered.

VIII. CONCLUSION

This paper presents a state-of-the-art review of meta-scheduling related technologies. We have presented a discussion of the inter-cloud needs and requirements and an analysis of meta-schedulers. Then we have discussed why meta-scheduling works in parallel to our research aim (scheduling in high dynamic settings) and we have presented a state-of-the-art literature review of various approaches main functionality as well as an evaluation of them on the basis of inter-cloud. The approaches presented herein will recognize the needs which eventually will lead to the modelling of a novel meta-scheduling algorithm for inter-clouds. The medium and high candidate works will be used as the fundamental aspects for considering already useful functionalities (e.g. history records, real time nodes' responses) and for fulfilling research gaps (heterogeneity using virtualization, rescheduling) towards inter-cloud meta-schedulers.

IX. 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] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] GICTF White Paper , 2010 Global Inter-Cloud Technology Forum, Use Cases and Functional Requirements for Inter-Cloud Computing, August 9, 2010, Available at: http://www.gictf.jp/doc/GICTF_Whitepaper_20100809.pdf, Accessed at: 02/07/2011.

[12] 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.

[13] Xu, B., Zhao, C., Hu, E., 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.

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

[15] Korkhov, V. V., Moscicki, J. T., Krzhizhanovskaya, V., V., Dynamic workload balancing of parallel applications with user-level scheduling on the Grid, Future Generation Computer Systems, Volume 25, Issue 1, January 2009, Pages 28-34, ISSN 0167-739X.

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

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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, pages542-550.

[25] Xhafa, F., and Abraham, A., Computational models and heuristic methods for Grid scheduling problems, Future Generation Computer Systems, Volume 26, Issue 4, April 2010, Pages 608-621, ISSN 0167-739X.