Decentralized Meta-brokers for Inter-Cloud: Modeling brokering coordinators for interoperable resource management

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

***Abstract*—The management of internal resources in large-scale environments is a crucial challenge due to the large number of users and service requests. In clouds, an efficient resource manager orchestrates internal resources by assigning brokers to users for acting on behalf of their clients. This is to map user requests to cloud datacenters for service allocation and execution. However, as cloud computing matures, it is crucial to enable the concept of inter-clouds, that is to say, enabling the collaboration and thus, the interoperation between several disperse (and highly likely heterogeneous) clouds. To this extend, we introduce the meta-broker concept for inter-cloud settings by modeling its conception in a total decentralized fashion. This is to coordinate different clouds brokers for establishing a reactive cross-exchange and service automation while offering transparency to users. We simulate an inter-cloud for measuring the performance of the average execution time for various users that submit concurrently a massive amount of services. The results show effective performance levels when operating under meta-brokering solution.**

Keqwords:* *Cloud computing, inter-cloud, inter-cloud resource management, Decentralized cloud brokers, inter-cloud meta-brokers.

I. Introduction

Over the recent years, the cloud computing has been emerged as one of the most important solution for delivering IT-oriented services to users. This is a new way of distributing services among resource consumers and providers by identifying consumers needs and sandboxing their requirements in virtualized settings. In this way, a high diversity can be achieved as different kinds of service could be deployed including software, hardware and infrastructure. Therefore, clouds offer significant advantages in remote propagation of services including flexibility, elasticity and automation in reduced cost. Most of the existing efforts [11] conclude to a narrow view of clouds by orchestrating settings as enterprise servers, clusters of hosts, or single datacenters with no inter-coordination capabilities.

In this work, we take a more inclusive view in which services could be encapsulated and distributed in a network of collaborated clouds for achieving a common goal [9], [5]. In practice, it is necessary for clouds to allow the transferring of services from one location to another in case of a user request. In advance, an interoperable cloud is the means to aggregate the different individual capabilities for achieving improved efficient utilization ranks. In this way, the overall service provision could be optimized while the users will not be aware of the collaborated infrastructure that they use of. This vision aims to the actual service rather than the infrastructure, thus shifting the focus on how to orchestrate a wider cloud service distribution. The interoperable cloud could achieve this; namely as inter-cloud that allows automatic service dissemination among collaborated resources.

Like clusters, cloud computing utilizes a centralized topology. In contrast, inter-clouds are more complex and somehow, it shares an identical distributed topology like other large-scale and dynamic infrastructures such as grids. These systems organize resources from multiple administrative domains to a single aggregated view in order to address common aims [8]. We vision an inter-cloud that encompasses resources from various e-infrastructures that may enter and leave reactively. This dynamic-ness denotes that real-time coordination is essential in decentralized interoperable collaborations. In this environment, massive computing capacity resides at a remote space and could be delivered in the form of software and/or hardware. These offered services are identical to job submissions that have been encapsulated in application execution requests posed by the end-users.

Specifically, the users submit their requests in a broker that communicates and monitors the whole service exchanging procedure [7]. This component is responsible for autonomous decisions by selecting a datacenter for forwarding the requests [6]. Then, each request is sandboxed in a virtual machine (VM) that satisfies user defined requirements. Various criteria are implemented in this level that include the demanded quality of service levels e.g. pricing, homogeneity in terms of hardware and software, and generic specification of the cloud hosts and VMs. These are enclosed in service level agreements (SLAs) that formally define the contract level of agreed terms between provider and client. Usually, this is related with the required computational power (performance) and run-time constraints. Further to this, the system involves a centralized brokering topology in which the broker has a complete knowledge of the datacenter configuration and parameters including hosts, VMs and utilization and allocation policies.

In this work, we present the meta-broker, a novel component that is placed on the top of each broker. By using cloud meta-brokers an inter-cloud is formed into an autonomously manage setting of interconnected sub-clouds. Current efforts in this direction organize (meta-) centralized topologies of brokers [3], [6], so various drawbacks derived from this narrow view. Herein, the work is inspired from the meta-computing concept and presents the model of a decentralized meta-broker. This is for establishing connectivity by managing virtualized resources of the interconnected infrastructure during service exchanging.

With this in mind, section II presents the related works in the area of meta-brokers in large-scale systems. The rest of the paper is organized as follows; section III illustrates the cloud and inter-cloud service-exchanging model, with the aim of presenting their architectural characteristics. Section IV presents the meta-brokering model along for addressing interconnectivity. Section V illustrates the experimental configuration and the discussion of the selected benchmarks when applied into a cloud and an inter-cloud setting. This also includes, a critical discussion of the performance of the simulation and the comparison of results. Finally, section VI concludes by presenting the further research steps.

II. Related works

The inter-cloud term denotes an interoperable environment in which various settings collaborate for purposely satisfying the quality of service (QoS) levels. This happens by extending the service distribution in the lower level of the infrastructure. Recently, various cloud vendors aimed to this joined cloud effort by establishing federations of clouds. It is noticeable that their state-of-the-art efforts have led to instituting collaborated clouds with joint initiatives. [4]. However, this vendor-oriented endeavour of inter-cloud has a specific control plane rather than a setting that it is based on future standards and open interfaces, which are available to be shared in the academic community. In addition, knowledge sharing, experimentation and testing within their systems have been limited to the wide range of researchers. In contrast to aforementioned work, the vision of inter-cloud as an inter-cooperative infrastructure has been introduced by [11], yet from a federated perspective.

[3] present a utility-oriented federation of various cloud computing environments. They conclude to a business model of system architecture including the most important elements (requirements) of inter-cloud in terms of complete system components along with the broker. The last one acts on behalf of the user that requests for service execution in centralized topology based on service level agreements (SLA) necessities. This change for every user and they are based upon their current requirements, e.g. pricing, accounting, VM distribution and configuration, and physical machine computational needs. [7] discuss that the broker acts as an SLA resource allocator by combining components to achieve the agreed benchmark among users and providers. This is a generic view of brokers that generate questions on how to manage the most effective resource allocation and scheduling. Principally, this has been addressed by considering different mechanisms e.g. advanced reservation strategies for guaranteeing service for users.

Initial efforts in this area are related with grid computing in which brokers assumed to be connected by high-speed networks [7]. Specifically, this could produce significant problems when a large bulk of user requests could cause a system bottleneck. This problem is identical with clouds, in which a new model is required to bridge the gap of resource selection, allocation and scheduling. To this extend [12] present a cloud broker for guaranteeing data transmission in cloud computing by meeting the users QoS requirements. Specifically, they present an experimental scenario to optimize data transfer within a cloud setting. Despite the fact that their selection algorithm increases network resource utilization, this solution aims to a service-oriented broker that satisfies more requests that does not include the perspective of brokering collaboration. Herein, we focus on the inter-cloud environments and brokers and meta-brokers in such systems. In particular, three works have been identified that aim to design (meta-) brokers for inter-clouds and these are presented bellow.

The first work of [11] discusses the brokering strategy in inter-cloud by presenting a centralized topology of a single broker. Specifically, in a multi-provision setting, a broker compares the SLAs of each provider and selects the most appropriate one on behalf of the user. This implies that the broker requires having a complete knowledge of the whole infrastructure along with current availability and communication quality levels. This centralized framework could be proven to be effective for small scale clouds (e.g. cluster-based), however, when it is extended in large-scale, it will face problems, e.g. single point of failure, bottleneck etc. In addition, the authors present their conceptual framework without presenting any experimental analysis.

In contrast to the aforementioned work, [3] present an inter-cloud datacentre broker for service execution. Specifically, in their setting the broker represents the user that submits the services in a cloud datacentre that is responsible for coordination with other datacentres for service exchanging though a cloud coordinator component. This happens by utilizing a component that dynamically sensing the availability of resources to interconnected clouds. In this work, it is suggested that each datacentre is responsible for coordinating with each other while the broker is only responsible for submitting and monitoring user jobs. At a first glance, this solution overcomes the centralized disadvantages of [11] as there is one coordinator for each datacentre. However, in the case of multiple users, thus multiple brokers, the system is transformed to a centralized setting. This means that one datacentre coordinator is responsible for managing all brokers along with their requests.

Work of [6] illustrates a federated cloud management architecture that facilitates autonomous behaviour. Specifically, they suggest an architecture that incorporates the meta-brokering concept for allowing transparent service execution. In addition, the meta-brokers interconnect with other brokers in order to aggregate the cloud capability. However, the federated cloud management architecture contains one component that is centralized generic meta-brokering service that orchestrates various brokers collaborations. The conceptual model does not provide any further technical and experimental discussion.

In contrast to all the above works, we aim to design a total decentralized meta-broker based on our previous inter-cloud model presented in [1] and [2]. For this purpose, the study extends the broker functionality by adding a meta-broker on top of the traditional broker for allowing communication with other meta-brokers during service submission. This means that throughout a request for service execution a meta-broker collaborates directly with other meat-brokers similar to a meta-scheduling system. This will offer significant advantages, as it will support highly interoperability, flexibility and heterogeneity while at the same time a job execution in a decentralized fashion. Several parameters are realized in this level e.g. pricing, homogeneity in terms of hardware and software, and generic configuration of the service and the VM. The next section presents the cloud and inter-cloud architectural issues in order to define the internal components and the requirements of the decentralized meta-broker that we utilize to design our proposed model.

III. Cloud and Inter-Cloud: Service Exchanging Phases

This section presents the conceptual architecture of the service exchanging-phase that occurs in a) a typical cloud-computing environment and b) a future inter-cloud setting of a multi-cloud participation. As exchanging phase we determine the stage in which services are submitted from the users to a cloud for service execution. Both cases illustrate the most important components from the scope of service delegation from users to the (inter) cloud.

A. The cloud exchanging architecture

In general there are various types of clouds, e.g. public, private, virtual, hybrid etc. and all types are directly related with the approach that provisions services to the end-users. For example in a public cloud providers use the public Internet for service delivery, while in private cloud providers deploy services within a single organizational domain. However, in each of the cases, the generic architecture includes the same functional operations described as follows.

The cloud core functionality is contained within a cloud datacentre that is responsible for the supervision of the end-user service requests. In addition, a cloud datacentre is in control of the hosts, which represent the lower level of the infrastructure. In fact, hosts (namely also as nodes) are the physical machines that contain the computational power. In real world, each physical host has been configured to the software level, to contain the core middleware functionalities. This is to say that an operating system has been installed along with hypervisor software for the deployment and management of VMs. In addition, appropriate software has been installed within the hosts for monitoring, metering and negotiating of various SLAs. The last one is part of the generic service contract that controls the formal service levels. This is to monitor the delivery of services by achieving time and performance constraints agreed among users and providers. All these constitute the basic physical components of a typical cloud by representing the low-level infrastructure. During the exchanging phase a user interacts with a broker for requesting service executions.

The broker represents the component that acts on behalf of the user and requests from the cloud system specific resources based on the contacted SLAs. Within the cloud setting, a management system offers the operational and business functionalities for responding to the user request. Theoretically, various processes take place within both components e.g. operational management that involves security control, fault tolerance management, and eventually scheduling and coordination. The operational management is also responsible for the core middleware functionalities including VM orchestration via the hypervisor. The business functionalities, on the other hand, include the SLA communication process involving payments and debts, which are decided prior to the service submission and scheduling. All the aforementioned parameters are handled by the datacentre and passed to the broker. Figure 1 illustrates the basic exchanging phase in which a broker a user submits a service request to the broker. The last collaborates with the datacentre operational and business component for finally sandboxing the service to a VM within datacentre host.

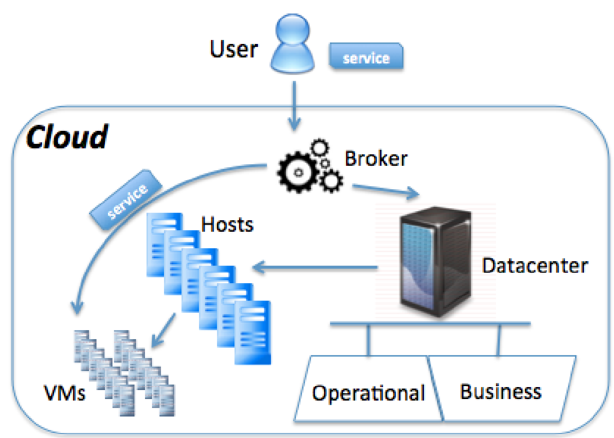

Fig. 1: The traditional cloud service-exchanging phase.

In principle, the broker operation includes a) the collection of the service from the broker, b) the collaboration with the datacentre operational and business component for processing SLAs and user specifications c) the submission of the service to datacentre, d) the monitoring of the service life-cycle within the datacentre modules (hosts and VMs), e) the monitoring of the service execution submitted from the user within the VM and f) the recording of data, generated during the whole cloud life-cycle, for being utlized in future as history logs of succesful job delegations. Thus, we conclude that the cloud broker offers the basic functionalities of the submission phase. It should be mentioned that further resource allocation mechanism (e.g. resource discovery, scheduling policies) executed within the datacentre should be assumed that are not related with the brokering function. The next section illustrates the inter-cloud service exchange architecture for exploring the brokering and meta-brokering operations.

B. The InterCloud exchanging architecture

The inter-cloud as a notion has been initially described by various works e.g. [11], [5] and [3] and all conclude that the inter-cloud is a metaphor for an interoperable setting of multi-infrastructures. Specifically, an infrastructure could be part of an inter-cloud if operates identical to a cloud. This includes a broker, a datacentre and/or cluster what could be virtualized, and an operational and business management component. Thus, the low infrastructure could be finally transformed to a cloud setting, thus constituting a part of the inter-cloud.

Architecturally, the inter-cloud shares common characteristics with a typical cloud setting as presented in section A. The difference among them is a need for a coordinator component to allow resources to be interconnected. This coordinator presented in [3] dynamically identifies the availability of resources from collaborated datacentres regarding their host level of utilization. Figure 2 shows the inter-cloud exchange model, which contains the cloud exchange system, along with the cloud components (user, broker, operation and business component).

Each time a consumer requests for a service, the cloud broker negotiates with the cloud coordinator for required resources, and related SLAs (e.g. agreed costs and computational power etc.). Then the cloud coordinator of the datacentre that has initially participated in service submission interacts with each cloud coordinator of interconnected datacentres. This is to say that the intra-components (operation and business) cooperate with the inter-component (coordinator) for interchanging services. In a similar vein, various clouds interact with other clouds coordinators and so on. It should be mentioned that the coordinators are totally decentralized in respect with the datacentre (one coordinator per datacentre). Yet, coordinators are centralized with respect to the actual cloud brokers (one coordinator per broker set of users). For each user the cloud generates a unique broker to handle the user-oriented requirements (SLAs, VM configuration etc.).

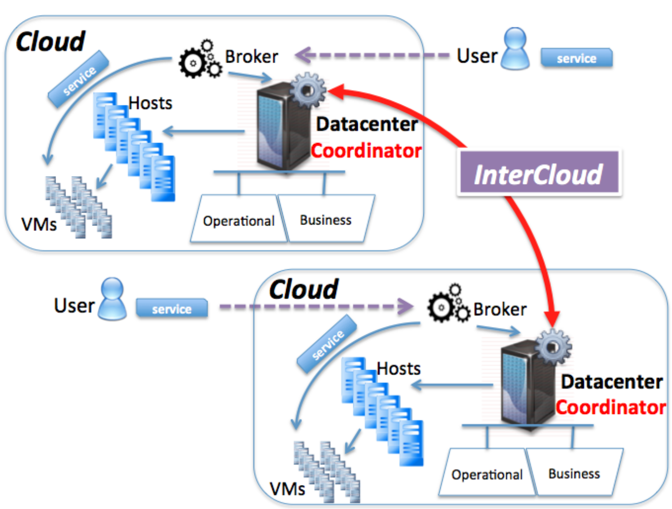

Fig. 2: The inter-cloud service-exchanging phase.

At last, when a service is accepted for execution, the cloud coordinator decides whether its own servers could complete the request, or it will be published as an offer to a different coordinator. Traditionally, this is achieved by flooding a message for resource availability and QoS specification (SLAs, hosts, VMs etc.) to each of the collaborated datacentres. Finally, the interconnected coordinators decide whether resources could be offered or not. This solution offers significant advantages over the non-interoperable clouds by enhancing the ability to offer a wider service distribution and placement but also, rise questions about the discovery, message passing and resource allocation way. In addition, cloud providers could easily integrate new datacentres for optimizing their computational power. This includes the migration of VMs and/or services among clouds for improving the fault tolerance mechanism by workload sharing.

However, as discussed previously the centralization nature of the coordinator in respect with the brokers could lead to a communication (service request and submission) bottleneck in the coordinator level since all the brokers of a cloud need to interact with a unique datacentre coordinator. In the case that a cloud has more than one datacentres, then a coordinator is assigned per datacentre thus pointing to the same concern. Furthermore, the centralized topology is correlated with a central point of failure, thus minimizing the fault tolerance of the whole system is a high requirement. This implies that if a coordinator goes down the whole communication among all brokers and coordinator breaks. Lastly, a common problem in such cases is the message passing among brokers and coordinator during service submission and execution. Massive amount of messages need to be passed by a single coordinator during run-time for keeping brokers updated with related information due to the requirement of continually broker monitoring.

In order to address the aforementioned drawbacks, we present a meta-broker inspired from the meta-scheduling paradigm. This is to optimize the submitted workload by combining multiple brokers into a single aggregated view, identical to distributed resource managers. We design the inter-cloud meta-broker to be totally decentralized and dynamic by enhancing the decision making process for service distribution. Next we present the model of the inter-cloud meta-broker.

IV. The meta-brokering model

The proposed meta-brokers are generated by the datacenter that represents the actual infrastructure (physical hosts). For each user the datacenter binds a broker that is responsible for controlling its requests, SLAs and monitor the whole service allocation and execution. Thus, the meta-broker provides an autonomous orchestrator that characterizes the initial point of the cloud from the view of the users. In addition, we assume that each meta-broker has a complete knowledge of the actual cloud infrastructure (e.g. datacenter characteristics, Hosts, VMs) as the expectation is that during job submission a meta-broker communicates with the cloud broker for information exchanging. However, the meta-brokers can have a complete or partial knowledge of other meta-brokers (related with the specific scenario) during service run-time. This perspective offers a high transparency level for the entire cloud since the users are only mapped to their assigned meta-broker, while the last one spontaneously directs the process.

Figure 3 illustrates the meta-brokering environment of two clouds. Specifically, the setting illustrates two users that request for service execution from their cloud providers. The cloud datacenter contains the normal broker (namely as local broker) that has a complete knowledge of the cloud infrastructure. In addition, the meta-broker is placed on the top of the local broker for controlling directly the user requests. For every user (and their services) the datacenter generates a new meta-broker that directly collaborates with the local and other meta-brokers of the system. Different from the existing solutions this study realizes inter-cloud by using meta-brokering operations. This moves the complexity of handling service requests from the datacenter level to a component that is harnessed to the actual service submission. In this way, meta-brokers could identify available resources more easily and reactively check for service execution opportunities. In addition, the meta-broker transforms the cloud to an inter-cloud, as it is able to communicate with other resource providers that offer better computational and/or market prospects.

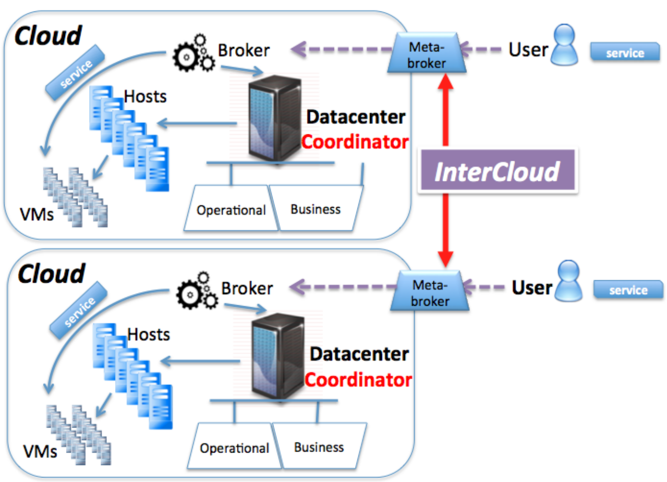

Fig. 3: The inter-cloud meta-brokering service-exchanging.

Nevertheless, this solution emerges questions about message passing, and information exposing among meta-brokers, along with security and trust in the level of cloud and inter-cloud. These shortcomings could be minimized or even eliminated by incorporating additional components for optimizing operations. For example, in the case of security we could implement a shared key authentication process for fundamentally gain secure access among clouds. In addition, by minimizing the amount of information exchanging could reduce the exposing of internal information. These are generic challenges that might enhance the significance of our model, yet currently are out of the scope of this study.

Technically, during the service submission, the meta-broker communicates with the local broker for information exchanging. This includes a total view of the local knowledge. In addition the meta-broker collaborates with other meta-brokers based on a specification selection of the cloud administrator. This fundamentally includes the swapping of keys recognizing each other for enhancing the security and trust level. Then each coordinator replies with an acknowledgement message for denoting the acceptance on collaboration. Then the requester meta-broker asks responder for information based on various criteria (e.g. resource availability, heterogeneity etc.). The responder meta-broker sends the information back by executing a matching procedure.

Relevant information that could be exchanged is a list of datacenter hosts, VMs, jobs (cloudlets) along with the characteristics of the datacenter (e.g. hosts CPU, Memory, number of processing elements (PEs), architecture etc.). Nevertheless, depending on the actual scenario information exchanging could be minimized to the level of resource availability or maximized to the complete internal knowledge analogous to the desired scenario. As the whole environment is based on the contacted meta-brokers real-time responses this minimizes the overall information exposition. It should be mentioned that service submissions, can be monitored from an external component and data could be utilized in future for enriching service submissions decision-making.

Next we present the experimental analysis and the simulation specification of an inter-cloud setting. Specifically, we implement our solution in the Cloudsim framework [4] for testing a simulated cloud computing infrastructures. Particularly, we present two scenarios; one simulates a typical cloud performance for certain service submissions and secondly an inter-cloud service submission that functions based on the meta-brokering solution.

V. The Experimental Configuration and Discussion

This section covers the cloud and inter-cloud experimental configuration, the simulation scenarios, the metrics to be used as benchmarks and the discussion of the testing results. Specifically, we integrate our conception in two case scenarios based on the same specification on the number of the users, brokers datacenters, hosts, VMs, bandwidth speed shared among components and cloudlets submitted in the system. In addition we utilize simple allocation policies of VMs instantiations in hosts and cloudlets in VMs (time sharing submissions).

As benchmark metric we use the average execution time of the cloudlet set that is calculated by the sum of the cloudlets execution time divided by the total number of cloudlets as given by formulae (1).

$$\text{AvgExecTime(cloudletset)} = \sum_{set=1}^{n} ExecTime[i]/Count[i] \quad (1)$$

In addition, hosts and VMs allocation happens in a time-sharing policy of first come first served fashion (FCFS). This indicates that at any given time multiple cloudlets could be allocated within the cores of a VM or a host depending the availability of the resource and the required computational power. With respect to the utilization model, cloudlets are submitted to VMs whenever a resource is available. Table 1 contains the specification values of the aforementioned parameters.

TABLE I: THE VALUES OF THE EXPERIMENT PARAMETERS.

Datacenter Hosts	VMs	Cloudlets
Datacenters: 3	VMs: 10	Cloudlets: 1000
Hosts: 2	Mips: 250	Mips: 1000
Mips: 1000	PEs: Single Core	PEs: Single Core
PEs 1st host: Quad Core	RAM: 512 MB	File size: 300 MB
PEs 2nd host: Dual Core	Size: 1 GB	Output size: 300 MB
RAM: 16 GB	Architecture: Xen	Allocation: FCFS
Storage: 1 TB	Bandwidth: 1 mbps	Utilization: Full
Bandwidth: 10 mbps	Allocation: FCFS	
Allocation: FCFS		

The two case scenarios aim of comparing the performance of the average execution time of a number of cloudlets submitted within the same inter-cloud environment. These scenarios are the following:

- An inter-cloud that produces a unique broker for each of the cloud users. Specifically, the cloud interconnection is based on a unique controller per cloud datacenter that is responsible for coordination. The forwarding service submissions are based on resource matchmaking upon service specification and resource availability. The controller is formed in a centralized topology with regards to the user brokers. This means that a single one is implemented for each datacenter on behalf of users as illustrated in figure 2.
- An inter-cloud that a) generates a unique broker for each of the cloud users and b) adds an extra meta-broker. This extends the standard broker functionality for forwarding requests to further interconnected meta-brokers. Similar to previous scenario, the service submissions are based on resource matchmaking and availability. The meta-broker is formed in a total decentralized topology that operates during the cloudlets submission as shown in figure 3 by being detached from the user.

It should be mentioned, that in both scenarios we assume that SLA matchmaking pre-exists along with resource matchmaking and availability. The SLA matchmaking functions prior to the cloudlets submissions at the broker level and being extended for simulation purposes at the controller and meta-broker. The actual experiment configuration integrates an inter-cloud setting to be composed by two sub-clouds. We specify both clouds to encompass 3 datacenters in total; the first cloud is comprised by two datacenters and the second cloud by one datacenter; thus three datacenters per inter-cloud. Each datacenter contains two hosts, and each host encompasses quad and dual core processing elements respectively. The million of instructions per second (mips) denotes the overall job execution capability of the resource either at a host or VM level. Correspondingly, cloudlets require a sufficient number of mips for performing executions.

In this setting we run an experiment of services submission that encompasses 1000 cloudlets in order to measure the performance of the inter-cloud. Each datacenter host creates 10 VMs for sandboxing service requests. For simulation purposes we integrate our solution in the Cloudsim framework that allows us to fully develop an identical environment and we utilize as benchmark metric the average execution time of the cloudlet set (1000).

We first simulate both scenarios using the same configuration to explore the behavior of the broker (scenario 1) and meta-broker (scenario 2) for various cloudlet workload submissions occurred by one user only. Primarily, several users concurrently submit a service that is comprised by a number of cloudlets into their broker (either the standard-scenario 1 or meta-broker-scenario 2). The initial intention is to explore the effectiveness of both scenarios when the number of cloudlets increases. Figure 4 demonstrates the average execution time of both cases, for one user submission and a variation of 100 to 1000 cloudlets. In addition, it shows the variation of the average execution time for exploring the trends of the average execution time for different cloudlets submissions.

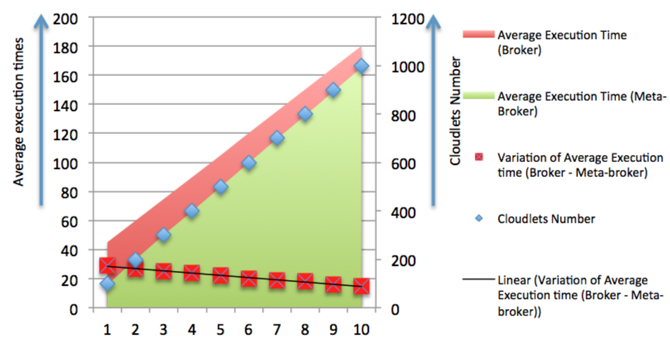

Fig. 4: The average execution times of broker and meta-broker, the variation of average execution times and the trend-line moving average) for cloudlets 100 to 1000.

We observe that for low workload submissions the broker of scenario 1 outperforms the meta-broker performance. However, as the number of submissions increases the difference of both scenarios falls significantly as the variation of the average execution time among brokers and meta-brokers decreases. This is shown from the trend-line of Figure 4. For extending our experiment we measure multiple users submissions for high workloads. This is particularly useful for clouds that offer collection of hosted application (e.g. Google Apps) in which multiple users request services concurrently. This implies that the development could be identical to a run-time submission in inter-cloud. In particular we measure the performance of 2 to 10 users that submit 1000 cloudlets within the inter-cloud.

Figure 5 demonstrates the performance of both scenarios for this set of users that submit 1000 cloudlets. The results show that the meta-broker (scenario 2) performance overtakes the standard brokering (scenario 1) by distributing the submission among 3 datacenters. Especially for high number of users the performance of meta-broker is significant lower. In addition, the trend-line (average moving) indicates that in cases of high workloads and number of users the variation of the performance (average execution time) between two cases is getting optimized for the benefit of the meta-broker.

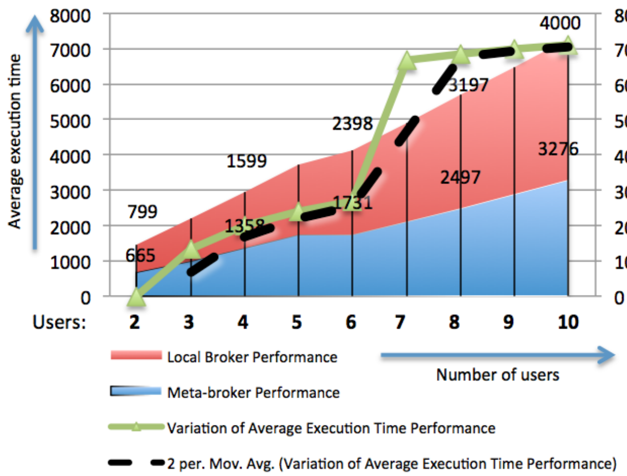

Fig. 5: The broker and meta-broker average execution time performance for 2 to 10 users submission of 1000 cloudlets in an inter-cloud of 3 datacenters.

To conclude, in this section we have presented that meta-brokering for inter-cloud offers significant advantages over the traditional model. By allowing collaboration at the meta-broker level we have optimized the large-scale submissions of multiple users by interconnecting multiple brokers into a distributed resource exchanging system. It should be mentioned that for a low number of users (e.g. one user of figure 4) the meta-broker consumes more time to execute cloudlets. Yet, for an inter-cloud this is not a realistic circumstance due to the large number of concurrent users that requests many service submissions. Our model simplifies the operation of the centralized interoperable datacentre-controller by moving the decision making process in the meta-broker level.

VI. Conclusion

This work aims to model the meta-brokering solution for inter-cloud. Existing efforts organize brokers in a (meta-) centralized topology, therefore various drawbacks derived from this perspective, like central point of failure and bottleneck in concurrent requests. For addressing these, we model a total decentralized component that is positioned on top of each traditional broker for achieving interoperation among clouds namely as inter-cloud meta-broker. The purpose is to distribute the whole setting and allow meta-brokers to collaborate with its other for trading SLAs and resources. By using this model the inter-cloud is transformed into an autonomously manage setting of interconnected sub-clouds. This moves the complexity of inter-coordination from datacentres to meta-brokers; thus achieving a decoupling of users from datacentres. We further design the inter-cloud meta-broker to be total decentralized, decoupled and dynamic by enhancing the decision making process for resource allocation and execution during cloudlet submission. The experimental analysis shows that our model outperforms the standard broker when the system encompasses high number of concurrent users and cloudlets submissions.

The future steps of this research include the further integration of dynamics e.g. VMs instantiation and an experimental analysis of the meta-broker for VMs migrations as introduced in [2], [9], [10]. Also, we intend to implement various scenarios to control reactive decisions e.g. failures of meta-brokers along with assumptions regarding different knowledge levels (complete or partial) of the actual meta-brokers pool. Finally, we plan to address a finer management of the scheduling decisions in both host and VM levels by implementing heuristic criteria and algorithms for performing meta-scheduling enhancement in inter-cloud environments.

References

[1] N., Bessis, S. Sotiriadis, V. Cristea and F. Pop, *An architectural strategy for meta-scheduling in InterCloud*, In Proceedings of the Proceedings of 1st International Workshop on InterCloud and Collective Intelligence (iCCI-2012), in conjunction with the 26th IEEE International Conference on Advanced Information Networking and Applications (AINA 2012), Fukuoka, Japan, March 26-29, 2012 [To appear]

[2] N., Bessis, S. Sotiriadis, V. Cristea and F. Pop, emphMeta-scheduling Issues in Interoperable HPCs, Grids and Clouds, International Journal of Web and Grid Services, 2012, (8):2.

[3] R., Buyya, R., Ranjan and R. N., Calheiros, *InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services*, Algorithms and Architectures for Parallel Processing, Vol. 6081/2010, Issue: LNCS 6081, Springer, 2010, 13-31 pp.

[4] R., N., Calheiros, R., Ranjan, A., Beloglazov, C.A.F., De Rose, R. and Buyya, *CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms*, Softw. Pract. Exper, Vol.41, No.1, 2011, 23-50 pp.

[5] J., Carolan and S., Gaede, *Introduction to cloud computing architecture*, White paper 1st edition, Available online by Sun Microsystem inc., 2011

[6] A.C., Marosi, A., Kecskemeti, G., A. Kertesz, P., Kacsuk, *FCM: an Architecture for Integrating IaaS Cloud Systems*, In Proceedings of the Second International Conference on Cloud Computing, GRIDs, and Virtualization (Cloud Computing 2011), IARIA, Rome, Italy, 2011, 7-12 pp.

[7] I., Rodero, F., Guim, J.,, Corbalan, L., Fong and S.M., Sadjadi, *Grid broker selection strategies using aggregated resource information*, Future Generation Computer Systems, 2010, 72-86 pp.

[8] M., Li, and M., Hadjinicolaou, *Introducing grid computing to a masters course: the design rationale and practice*, In Proceedings of the 5th WSEAS/IASME international conference on Engineering education (EE'08), Jaime Lloret Mauri, Azami Zaharim, Andrei Kolyshkin, M. Hatziprokopiou, Athina Lazakidou, Michail Kalogiannakis, Konstantinos Siassiakos, and Nikos Bardis (Eds.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 2008, 150-154 pp.

[9] S., Sotiriadis, N., Bessis and N., Antonopoulos, N, *From meta-computing to interoperable infrastructures: A literature review of schedulers and simulators*, In Proceedings of theAdvanced Information Networking and Applications (AINA 2012), Fukuoka, Japan, March 2012 to appear.

[10] S., Sotiriadis, N., Bessis and N., Antonopoulos, N, *Towards InterCloud schedulers: A survey of meta-scheduling approaches*. In Proceedings of the6th International Conference on P2P, Parallel, Grid, cloud and Internet Computing (3PGCIC 2011), 2012, 59-66 pp.

[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: 01/03/2012

[12] Y., Yang, Y., Zhou, L., Liang, D., He, and Z., Sun, *A Sevice-Oriented Broker for Bulk Data Transfer in Cloud Computing*. In Proceedings of the 2010 Ninth International Conference on Grid and Cloud Computing (GCC '10), IEEE Computer Society, Washington, DC, USA, 2010, 264-269 pp.