


Article

Distributed Meta-Brokering P2P Overlay for Scheduling in Cloud Federation

Sajid Latif ^{1,*}, Syed Mushhad Gilani ¹, Rana Liaqat Ali ², Misbah Liaqat ³  and Kwang-Man Ko ⁴¹ University Institute of information Technology, Pir Mehr Ali Shah Arid Agriculture University Rawalpindi, Rawalpindi 46300, Pakistan² Department of Physics, COMSATS University of Information Technology, Islamabad 44000, Pakistan³ Department of Computer Sciences, Air University Islamabad, Islamabad 44000, Pakistan⁴ Department of Computer Engineering, Sangji University, Usan-dong 26332, Korea

* Correspondence: sajid_latifs@yahoo.com

Received: 23 May 2019; Accepted: 29 July 2019; Published: 31 July 2019



Abstract: The interconnected cloud (Intercloud) federation is an emerging paradigm that revolutionizes the scalable service provision of geographically distributed resources. Large-scale distributed resources require well-coordinated and automated frameworks to facilitate service provision in a seamless and systematic manner. Unquestionably, standalone service providers must communicate and federate their cloud sites with other vendors to enable the infinite pooling of resources. The pooling of these resources provides uninterpretable services to increasingly growing cloud users more efficiently, and ensures an improved Service Level Agreement (SLA). However, the research of Intercloud resource management is in its infancy. Therefore, standard interfaces, protocols, and uniform architectural components need to be developed for seamless interaction among federated clouds. In this study, we propose a distributed meta-brokering-enabled scheduling framework for provision of user application services in the federated cloud environment. Modularized architecture of the proposed system with uniform configuration in participating resource sites orchestrate the critical operations of resource management effectively, and form the federation schema. Overlaid meta-brokering instances are implemented on the top of local resource brokers to keep the global functionality isolated. These instances in overlay topology communicate in a P2P manner to maintain decentralization, high scalability, and load manageability. The proposed framework has been implemented and evaluated by extending the Java-based CloudSim 3.0.3 simulation application programming interfaces (APIs). The presented results validate the proposed model and its efficiency to facilitate user application execution with the desired QoS parameters.

Keywords: scalability; meta-brokering; local-brokering; federation; interconnected

1. Introduction

Interconnected cloud (Intercloud) is similar to the Internet in that the latter has been developed by interlinking multiple standalone networks. Therefore, in the same paradigm, the evolution of the Intercloud is quite intuitive. The interconnection of geographically distributed resources through uniform standards and protocols is more resilient and provides elasticity in peak time to maintain Service Level Agreement (SLA) [1,2]. Due to dynamicity in workload and resources states, client requests are serviced transparently in this paradigm [3–5]. An Intercloud [6–10] architecture, which connects different service providers from various locations to reduce the probability of service blockage is indispensable in the context of SLA improvement and seamless service provision. Therefore, collaboration and communication among geographically disparate clouds is required to overcome the limitations of single, standalone service providers [11–13]. Apparently, under peak

load, single providers are unable to meet application needs due to exponentially growing cloud user services. Apart from application needs, small- or medium-level providers must federate their resources to compete with big vendors in terms of market perspectives. The Intercloud is a paradigm to provide a large-scale infrastructure that facilitates just-in-time, opportunistic, and scalable provision of application services. The primary focus of this paradigm is to achieve QoS targets consistently under varying workload, resources, and network conditions. Intercloud methodology enhances client satisfaction, and causes a reduction in the violation of SLAs at runtime [14]. Different scenarios of interconnecting clouds are depicted in Figure 1, which improves various schemes. However, this study only focuses on a federation paradigm wherein providers interconnect the resources voluntarily. The words meta-brokering components, instances, and peers are used interchangeably in the same context.

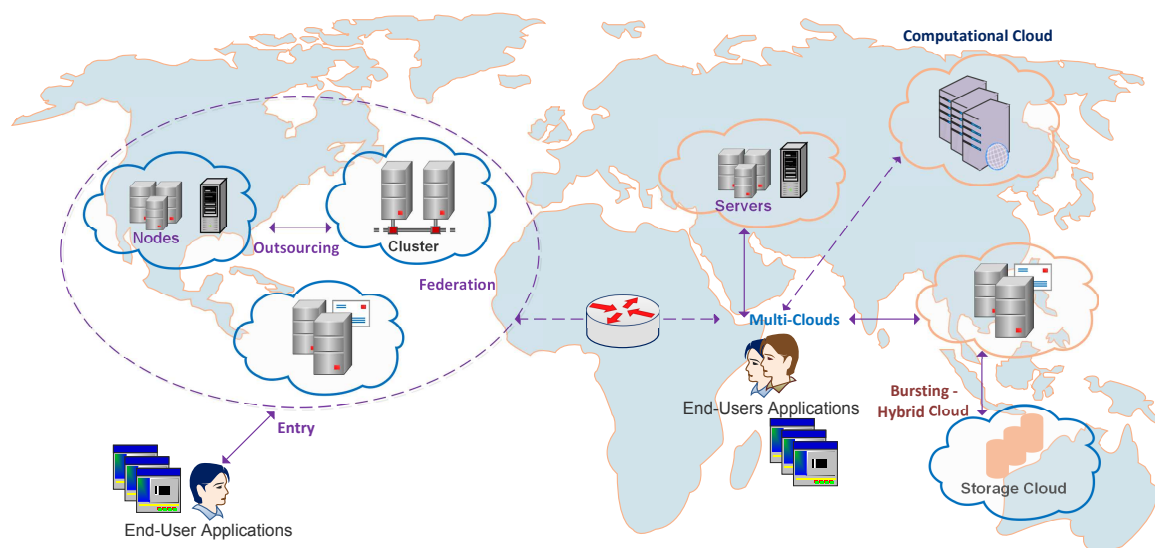


Figure 1. Interconnected cloud scenarios.

1.1. Brokerage Topology Paradigm

Brokerage architecture for the formation of the Intercloud federation comprises various topologies, e.g., centralized, decentralized, and hierarchical [15–17]. A centralized scheme is a single meta-scheduler- or broker-centric where all clouds are controlled through a core component [18]. It also allows the interface to submit application requests for various computing resources treated as middleware between users and service providers. In a decentralized approach, brokering components are distributed that represent their associated cloud, and realize a federation by inter-broker communication. Collaboration among these components also facilitates a discovery process in a seamless manner. Moreover, in this scheme, user requests are submitted to each meta-broker, where, after analyzing the states of resources of each service provider, requests are diverted to more optimal datacenters through their respective local-brokers. Local-brokers have ownership of the resources of datacenters, and perform actual allocation by using various algorithms. Therefore, local-brokers maintain constant interaction with meta-brokers as resource use and availability changes dynamically. Hence, in view of highly dynamic scenarios, meta-brokering components cannot make prior decisions without receiving updates on the consumption state resources.

In the Hierarchical approach, the broker connects to various clouds, and that broker communicates with other brokers to search other clouds whose metrics match the criteria of clouds that need to create a federation. The discovery of remote peers (meta-brokers in our model) in the decentralized mode is an essential factor for exchanging services and workload information, so that functionality could be performed in a distributed manner by each peer of the federation mandatorily and same is carried out

by single entity (the broker) in the centralized mode. In the proposed approach, a distributed paradigm brokering components associated with each service provider is interconnected and forms the topology or peer overlay network. Organizing the structure of distributed brokers is a critical element, as queries and metric statistics need to propagate in an efficient way among peers (meta-brokers). Therefore, in the modeling phase of the decentralized federation structure [19,20], a systematic and automated approach is required for exchange of the essential metrics among peers regarding the associated resource states. Based on this critical information, meta-brokering peers perform the meta-scheduling of client applications, and manage the dynamicity of federated cloud sites. The promising features of the Intercloud paradigm that have been gaining momentum recently are listed below [21].

- To serve geographically distributed cloud customers, the interconnecting phenomena of a standalone service provider in a different location is an optimal and cost-effective solution in view of unpredictable incoming service requests.
- The interconnecting cloud paradigm is also a business trend where Cloud Service Providers (CSPs) can lease resources to other CSPs under underloaded resource conditions.
- The federation setup provides an infinite pool of resources that minimize the risk of SLA violation and improves confidence in enterprises using cloud technologies.
- CSPs can harness the Intercloud federation to provide a scalable service provision under varying workloads and handle a dynamically changing environment.

1.2. Research Focus and Contribution

The overwhelming conditions of standalone service providers are a major reason for service outage and SLA violation. Therefore, under such conditions, the moving of user applications into other resource domains is essential for ensuring QoS constraints. Resource domains belonging to different providers may be located in various locations. Therefore, the management of computing resources in large-scale distributed systems, e.g., the grid and Intercloud, is really challenging in terms of requiring uniform architectural designs, standards, and well-automated brokering techniques [22]. Scheduling approaches to manage the resources and user applications can be applied in decentralized or centralized modes in terms of computing scalability factor. A centralized scheme is appropriate for small- to medium-scale paradigms, but is not effective for large-scale ones, for certain significant reasons, e.g., less scalability, performance bottleneck, and a single point of failure.

Resource scheduling in a federation to allocate user application requests efficiently and avoid delayed responses is a complex process. Therefore, to achieve scheduling of dispersed resources in federated domains, a distributed brokering paradigm must be focused on parametric sweep applications (PSAs). These PSAs belong to different scientific domains e.g., high-energy physics (HEP), ad-hoc network simulation, and financial modeling, and are regarded as killer application models for the framing of high-throughput computing (HTC) applications. Meta-brokering is a top-level brokering methodology that performs global decisions and meta-scheduling operations among distributed domains in a coordinated manner. These instances function on top of existing resource brokers to manage elasticity and interoperability among heterogeneous sets of resources [23–27]. Communication among meta-brokering instances at a global level in a seamless manner sets up the federated formation.

These distributed instances in the proposed scheduling methodology maintains a data structure of updated information regarding the resource of associated cloud domains and behaves dynamically. These distributed instances accelerate an optimal resolution of service distribution through best-fit load-aware routing functionality of end-user applications. To adopt the best-fit load-aware meta-scheduling technique, distributed instances analyze the resource state information after periodically collating all the associated local resource management systems (local-brokers). Since, in large-scale computing, datacenters belong to different service providers that are geographically distributed, centralized meta-brokering is not appropriate for various reasons e.g., less scalability, less reliability, single point of failure, and performance bottleneck. Consequently, to curb these issues, decentralized brokerage design is mandatory for dynamic cross-cloud service provision and efficient

scaling of the workload. These meta-brokering instances in peer-to-peer overlay networks exchange essential metrics based on data structure for operative decisions.

The proposed model ensures scalability and availability, and targets optimal datacenters to maximize the service throughput and ensure performance metrics. The main contribution of this work is listed below:

- The design, implementation, and evaluation of a distributed meta-brokering-driven federation paradigm for elastic service provision with scalability.
- The implementation of threshold-based load-aware routing in P2P overlay for PSAs under varying workload.
- A distributed monitoring mechanism to facilitate dynamic brokerage while updated peers through pub/sub-like patterns periodically.
- A honeybee-inspired resource allocation to local datacenters to minimize processing time of user applications.

The rest of this paper is organized as follows: work in related domains is discussed in Section 2. Section 3 describes the proposed overall system architecture with respect to related modules in various tiers and implementation strategies. The problem is formulated in Section 4, along with simulation results. Finally, Section 5 concludes the work and proposes further research avenues.

2. Work in Related Domains

Market-oriented initiatives for the Intercloud were taken by Melbourne University based on a centralized mechanism for service provision inspired by the semantics of inter-grids for resource exchange among providers [6,28]. Cloud Exchange (CEx) in this architecture is a centralized component that manages the collaboration with other providers through a coordinator component that is deployed on each provider. This architecture model can view the marketplace using the CEx component, where resources are sold (published) by other clouds, and needy application brokers can lease required resources. Hence, price-aware application brokering in this model can be realized. CEx pools the resource information in the registry to offer to others, and the cloud coordinator updates the status of the associated cloud resource in the CEx registry periodically. A coordinator for an extendable cloud is also discussed by [29].

Intercloud Meta-Scheduling (ICMS) developed by [30] is a dynamic service-provision model to achieve scalability across multiple clouds when there are spikes in demand. The ICMS model is based on meta-brokering components in a decentralized manner. In view of handling dynamism and other functional aspects of the cloud-servicing paradigm, the proposed architecture is more feasible for overcoming the issues of centralized counterparts. Implementation of this system is carried out in a toolkit (SimIC), which is based on the simjava simulation library. Optimization of job scheduling relies on the message exchange [31] pattern among distributed nodes. In this approach, multiple meta-brokering instances are generated against multiple users, one for each user for the whole service-provision cycle.

The Intercloud Resource-Provision System (IRPS) proposed by [32,33] is a semantically driven ontological model that facilitates application resource provision to distinct clouds. The main objective of incorporating the ontological notion is to realize interoperability among service providers with distinct resource policies. The proposed system forms a federated setup through the interaction of key components e.g., (broker, co-coordinator, Intercloud directory). The broker component selects the appropriate provider for the application submission, the cloud coordinator publishes the current states of the resources of the respective vendors through the Intercloud directory, and this component also updates the availability of resources dynamically. Each cloud site in the federation maintains a Semantic Metadata Repository (SMR), where the semantic descriptions of components are stored.

Ref. [3] proposed the Intercloud federation framework, wherein several standalone providers collaborate with each other seamlessly to maintain QoS and ensure SLA. Components in the model

interact dynamically to realize the federation process while ensuring QoS orientation. High-level architecture of the proposed model contains various components (e.g., cloud monitor, resource monitor, resource allocator, workload predictor). The monitoring of the resources is performed by a resource monitor periodically to check the availability of resources; it also intimates the Federation Negotiator (FN) in case of non-conformance of SLA parameters. The allocator is responsible for allocation of the available resources.

To overcome the interoperability issue raised due to the heterogeneity factor of resources among the service provider, Intercloud architecture must implemented by [34] using the OCCI standard to facilitate application migration in a seamless way regardless of the underlying resources semantics. The proposed generic architecture for Intercloud computing is broker-centric, which mediates the application execution cycle accomplished by using the latest open standards to realize interoperability. Broker entity with various modules operate in a centralized mode and locate the appropriate service provider to meet functional and nonfunctional SLA parameters.

Ref. [35] used the distributed brokering instances to interconnect geographically dispersed datacenters functioning in real-time scenarios instead of a simulation environment. Since this federation schema is configured in a real-time environment, there is the need to implement standard interfaces, uniform protocols, data formats, and compatible architectural components among service providers that facilitate collaborations for enabling cloud interoperability. The same is achieved in this research through common cloud brokering abstraction functioning in a distributed manner. Inter-broker communication in this closed-scope federated topology procures the resources and performs matchmaking based on derived metrics (load state of DCs) from datacenters under different scenarios e.g., (storage space, distributed availability, migration, and stability).

InterGrid Gateway(IGGs) components were proposed by [36] to be placed on the InterGrid, which facilitates cross-grid cooperation to set up peering topology among grids. IGGs are placed on top of each participating distributed grid that functions in a decentralized manner for efficient resource discovery. However, the non-availability of fault-tolerance in this architecture leads to non-communicated grid cooperation.

In [37], a delegated matchmaking approach is described for interconnecting several grids. This approach does not require a central control mechanism to managing the operation. Simulation results of this model demonstrate better administrative benefits and performance metrics. However, a heterogeneity issue is raised in distributed settings on a large scale. Table 1 presents the highlights of work in related domains.

Table 1. Highlights.

Techniques	Objectives	Achievements	Limitations
Intercloud [6]	Developing the market-oriented cloud federation model	SLA-based application brokering with QoS parameters. Model can be applied in public or private cloud mode.	Maybe less reliable due to centralized nature of CEX. Vulnerable in case of failure of central CEX.
ICMS [30]	Developing the interconnected cloud meta-scheduling	Large-scale dynamic job distribution and scheduling	Multiple generation of meta-brokers for multiple users create complexity and overhead
IRPS [32,33]	Ontology-based federation development	Interoperability through ontological notion among service providers	Does not provide large scalability due to single scheduler component
Federation Framework [3]	QoS-oriented cloud Federation	Maintain and ensure the SLA-based service provision-model	Federation Negotiator (FN) being sole component does not scale well in peak time
OCCI Driven Broker Federation [34]	Implementing the OCCI standard to make interoperable service provider	Conformance the Functional and Nonfunctional SLA metrics	Centralized brokering entity not suitable for flash crowd
Distributed Broker instances [35]	Configuration of Federated schema in real environment	Broker instance provide uniform interface and better scalability	Monolithic approach of distributed abstraction create management complexity
IGG [36]	Interlinking the separate Grids by peering arrangement for collaboration.	Improve the average response time	Brokers are self-interested and not global.
DMM [37]	User satisfaction by temporarily binding local resource with remote through DMM	Reduce the administrative overhead and enhance the performance	System dynamicity is not considered and steady state is assumed. Heterogeneity issue is not resolved.

Considering the merits and limitations of large-scale settings that have been discussed above, the proposed approach is more generalized, which functions in a distributed manner and is comprised of distinct modules. This modularized approach increases system robustness and implies fine-grained management. The significant characteristic is the monitoring mechanism that facilitate the dynamic brokerage to counter the overwhelming and unpredictable conditions.

Semantic Web Techniques

The Semantic Web is an extension of the existing World Wide Web (WWW) with great potential to overcome issues e.g., interoperability and application portability. This mechanism uses various languages e.g., OWL and SPARQL, and OCCI standards. Researchers in the existing literature have predominantly focused on ontological models to describe cloud service semantics at various levels. In this regard, mOSAIC ontology allows interoperability and service description, which can also be leveraged to address interoperability issues in multiple clouds. Ontologies facilitate the representation of semantic information. In the Semantic Web, ontologies are used to specify a domain of interest and related concepts [38–40]. However, the focus of this study is on the brokering-driven scheduling in the cloud federation, in which service providers interconnect resources voluntarily.

3. Proposed System Design and Spectrum

It is obvious from the above discussion that the provision of optimal resources in a large-scale dynamic and decentralized environment is a challenging task in terms of meeting application requirements. Therefore, it is indispensable to design and develop reliable, scalable, and dynamically decisive systems. A system should be capable of discovering the optimal services in a shorter span for application execution in more competitive situations. To curb the issues of centralized counterparts e.g., single point of failure, scalability, and reliability in large-scale distributed and highly dynamic environments, in this study, a decentralized high-level meta-brokering-driven approach has been proposed. These components inter-communicate in a peer-to-peer (P2P) manner and constitute the overlay topology, enabling distributed brokerage. Decentralized meta-brokering components in this scheme make decisions at higher levels and manage dynamicity through interacting with other peers for auto-scaling of the workload. In the proposed model, meta-brokering instances realize interoperability [41–43] in terms of application and metric exchange among resource sites in a federation. Critical information exchange by these instances is enabled after authentication and peer processing at a global level. Meta-broker instances are interconnected in overlay topology and each instance maintains the local queue for incoming user requests to send to associated local-brokers or forward to remote peers under certain conditions. Figure 2 depicts the global view of the proposed system design.

3.1. Distributed Meta-Brokering Overlay Topology

Meta-brokering is the technique of brokering over resource brokers instead of resources. Meta-broker peers in overlay network communicate with each other in a P2P manner on the application layer for the exchange of essential metrics (e.g., load and applications). The P2P communication paradigm is more scalable than the client/server approach in large-scale distributed computing environments. For large-scale content distribution and sharing of resource information among distributed domains, P2P methodology is promising and flexible [44–47]. P2P systems with robust architectural design scale well to counter heavy user traffic with efficient resource use.

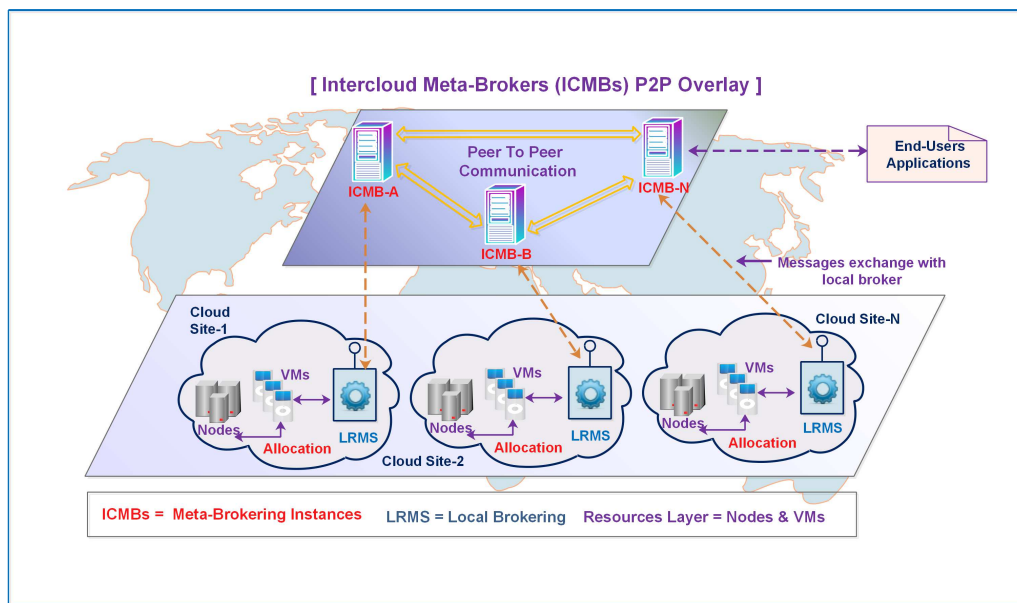


Figure 2. Global view of meta-brokering overlay for federation configuration.

In the proposed model, loosely coupled peers (meta-brokers) inter-communicate to provide more flexibility through asynchronous interaction, and maintain periodical updated statistics of each other, e.g., load metrics. P2P meta-brokering instances in overlay networks update each other regarding monitored information through pub/sub-like protocols, as they are powerful abstractions of the distributed nature of computing. This pub/sub-driven communication is more appropriate for highly dynamic systems because of its inherent properties, e.g., asynchronous, decoupling, temporal freedom, and overcoming the scalability issues of centralized system. In this model, each meta-broker instance corresponds to the associated local-broker for sending user applications, and receives the dynamic resource state. Each meta-broker instance uses the data structure of the dynamic metrics and finds other peers using logical identifiers for forwarding applications. The data structure of metadata is depicted in Figure 3. The salient factors of the decentralized approach are highlighted in Table 2.

Table 2. Salient features of P2P decentralized systems.

Factors	P2P-Driven Decentralized	Centralized Scheme
Single Point of failure (SPOF)	No	Yes
Performance bottleneck	No	Yes
Reliability	Good	Bad
Redundant information query	Great quantity	None
Scalability	Quite well	Bad
Traffic	Large	Small
Architecture	Peer-to-Peer	Client-Server

3.2. Decentralized Brokerage

Information exchange among meta-brokers (peers) in overlay networks realize the federation setup and facilitates decentralized brokerage. In this meta-broker overlay structure, information regarding service discovery is exchanged among users and local-brokering components through interaction with remote peers. Distributed meta-brokering instances provide interoperability in terms of job exchange and load scalability among federated sites. Meta-brokers communicate asynchronously with peer brokers to exchange resource state information for optimal decisions regarding application

forwarding; this substrate provides an elastic service provision seamlessly under overwhelming conditions of any site in the federation.

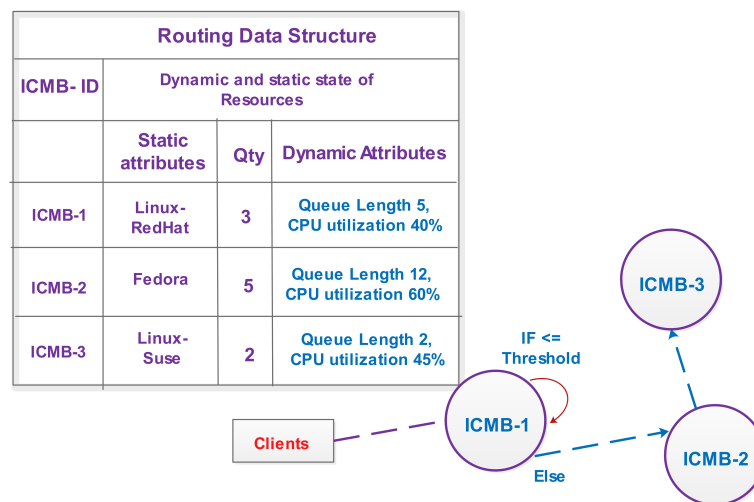


Figure 3. Meta-brokering level data structure.

3.3. Collaborating Paradigm in Overlay Topology

Initially, the distributed instances (meta-brokers) in overlay topology exchange the shared keys to establish the trust level and federation process. Overlay links are more flexible for building customized communication paradigms and protocols, e.g., information routing in a proposed system. Overlay methodology also supports avoidance of network congestion to adaptively select a shorter latency path. In the proposed system, each meta-broker component (peer) maintains a data structure of the other ICMBs in the federation to make an optimal decision regarding routing queries. This phenomenon controls the propagation of queries in a flooding manner, and sends queries only to more optimal peers in terms of load capacity. Application sequences among core entities of the federation after the authentication process among meta-brokering instances for peer arrangement is depicted in Figure 4.

The objectives of the meta-brokering notion in the proposed system are three-fold.

1. Retain the system decentralization and scale the workload in case of overwhelming the particular associated cloud.
2. Perform decision-making at a global level for application forwarding on remote sites, while exchanging essential metrics.
3. Share the load among peers in the federation and minimize the waiting time of the user applications.

The user submits queries with required attributes for job execution to associated meta-brokers in the overlay paradigm. Requests are served in a first-come-first-served (FCFS) manner in the assigned peer (ICMB) queue. The matchmaking component matches the required attributes and checks the predefined threshold values of all the queued running resources. After successful matching of the query parameter, the meta-broker sends results back to a particular client. In the case of exceeding a predefined threshold of resources, a queue request will be dispatched to other peers based on routing-table statistics. If more peers are available with the required attributes, then preference will be given to the peer with fewer or fewest loaded resources.

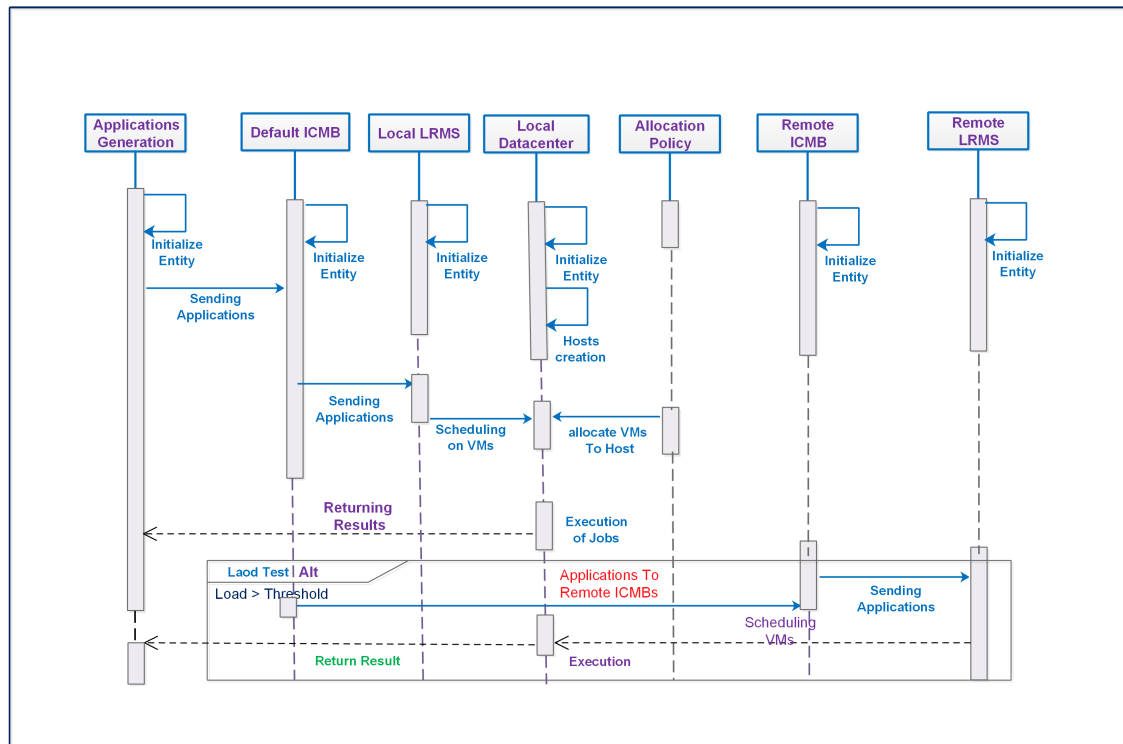


Figure 4. Application Sequence in Federated Entities.

User application forwarding in the proposed system proceeds through meta-brokering components, which map the user requests to the local-brokers of a specific domain. Decision-making for forwarding applications among meta-broker peers is based on time-bound statistics, which helps to avoid the overwhelming conditions of specific cloud sites. Queue length of the particular single resource (VM) indicates the load and collective load of all resources present in the overall load of any participating cloud domain. In the case of exceedance the overall load from certain thresholds, or the failure of required attributes matchmaking, subsequent applications will be routed towards optimal domains. Query routing in this approach does not flood in all directions, which saves bandwidth cost. Each peer (meta-broker) maintains the data structure of the updated dynamic information, which is retrieved through the monitoring module. The data structure associated with each peer is updated while exchanging metrics through pub/sub-like asynchronous protocols, as depicted in Figure 5. In this way, all meta-brokers get updated regarding the current system state, and can adopt a best-fit load-aware policy for sending applications. This method of information exchange among peers (meta-brokers) is critical for scaling workload towards optimal domains.

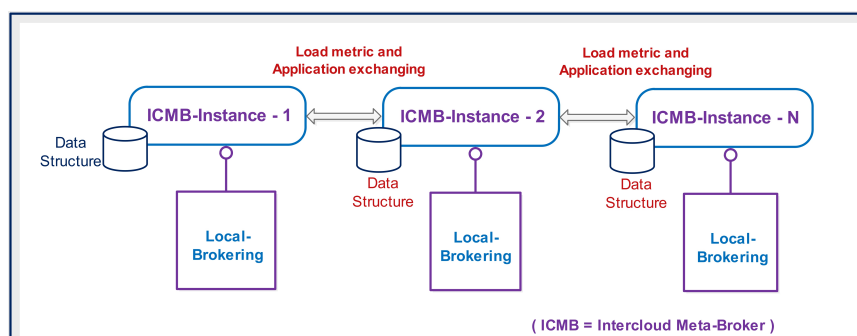


Figure 5. Meta-brokering information exchange.

3.4. Internal Information Flow

The scheduling of multiple user applications on cloud-federated resources is enabled through the interaction of meta-brokering components at the global level. Interaction flow is reflected in Figure 4, which occurs through messages passing among Intercloud meta-brokering (ICMBs) components. The Intercloud federation paradigm extends the capability of a single cloud in terms of achieving various resources as a service, which is distributed geographically at large scale. Therefore, in this design setting, the development and configuration of interoperable components for service exchange among various domains, and scheduling the virtual resources is a challenging task. Algorithm 1 describes the job-forwarding policy in the peer (meta-broker) overlay network.

Algorithm 1 For Application Forwarding (Pseudo-code).

```

Require:
R_ICMBR : The requester or default meta-broker (peer)
App_Queue(): Incoming application jobs that need
to be allocate locally or send remotely
ICMB_Map(): contain the current load of all participating
meta-brokers
Load_Threshold: store the value of load limit
dispatch_App(): send applications to remote peer or local-broker
L_ICMB_LBRK: Local-broker (LRMS) of default meta-broker (peer)
R_ICMB: Remote Intercloud meta-broker (ICMB)
timeInstance: hold the current system time

Begin
At: timeInstance
For All Each APPjob in App_queue() do
assign getID_LeastLoad_ICMBMAP() to remote_Icmb_ID
assign get_ICMB_Map(remote_Icmb_ID) to R_ICMB
IF default_ICMBR_LBRKload <= Load_Threshold
dispatch_App() to L_ICMB_LBRK
Else
dispatch_App() to R_ICMB
EndIf
EndFor
End

```

3.5. Modular Stack of The System

The meta-brokering-driven federated system is classified into distinct layers with respect to functionality. The high-level or meta-brokering layer performs the function of middleware between users and the rest of the cloud system. These components (meta-brokers) also interact with other peers at the same layer and exchange information about services. Critical information traverses the overlay topology to scale the load efficiently for optimal response. Next to this layer (meta-brokering) is the local resource management system (LRMS) or local-brokering, which maps user requests to virtual resources (VMs) according a predefined set of algorithms, and monitors the resource use states dynamically. The low-level or physical layer is the hardware infrastructure, consisting of physical and virtual hosts. The purpose of the division of the system functionality into different layers is to perform fine-grained management of the whole service-provision cycle in the federation paradigm. The modularizing approach simplifies the operation and does not affect it as a whole in the case of catastrophic failure, and improves system robustness. Meta-brokers provide a uniform interface for application submission and enable interoperability among distributed domains while communicating in a P2P manner.

3.6. Breaking Down System Functionality into Layers

The proposed model for service scheduling in the Intercloud is divided into three distinct layers—Resources Layer (RL), LRMS or local-brokering), and Global or Meta-Brokering Layer, as depicted in Figure 6. This architecture is formed into distinct layers to maintain the decoupled settings of different module functionalities for fine-grained orchestration. These components collaborate through message-passing to achieve a current snapshot of the overall system. The following points describe the core method of information collection and sharing in the proposed system.

- (1) Meta-brokering instances (peers) in overlay schema exchange the essential metrics of the associated realm in a periodic manner asynchronously. The exchange of such information is crucial to facilitate the dynamic brokerage at global level.
- (2) Collect time-bound statistics of resources through the local-broker's monitoring module on the layer below.

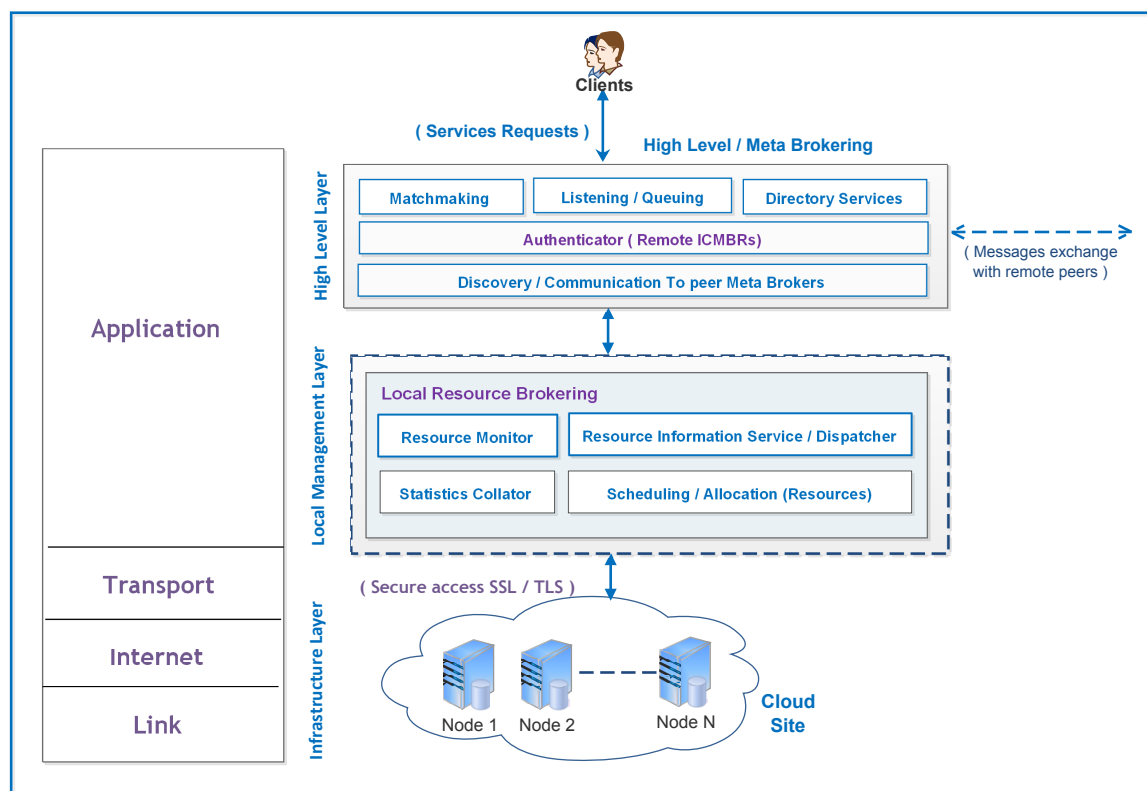


Figure 6. High-level design (federated site) mapping with TCP/IP layers.

3.6.1. Meta-Brokering Layer (MBRL)

Each meta-brokering component in the system retrieves the updated metrics of the associated cloud. These metrics are exchanged among them and are maintained in the form of data structure for decision-making. Considering this data structure, applications are routed towards optimal domains. Meta-brokering components (peers) function as clients as well as servers, e.g., asking for resources and serving the application requests of remote peers in particular conditions. These instances are constantly aware of each other regarding associated domain state, e.g., load. The discovery of services with required attributes in the proposed paradigm is carried out through inter-communication of meta-brokers, which also realize the federated environment. These components do not have autonomy of resources regarding the associated cloud site; therefore, they do not perform typical scheduling of user applications. However, they only receive dynamic statistics from a monitoring module of

the associated local-broker to keep the data structure updated with the latest information. Global decision-making takes place using these instances based on latest information.

- Exchange the shared keys to set up the trust level among peers (meta-brokers) in overlay topology.
- Meta-brokering service provides the uniform interface for user application submission.
- Analyze the load information sent by the local-broker stored in the CMDS for forwarding requests locally or remotely.
- Implement the load-aware meta-scheduling of user applications in overlay nodes.
- Exchange the dynamic information with other peers through pub/sub-like pattern e.g., load and average queue length, to realize the actual status of the cloud site in the federation.
- Maintain the system in a distributed mode and share the intersite workload.

3.6.2. Local Resource Management Layer (LRML)

Management and monitoring of local resources is performed on this layer by a local resource broker and its associated components.

- Communicate with resource information service (RIS) to get the available registered resources.
- Local-broker receives the application workload from the meta-broker of the above layer.
- Perform the resource allocation process of the user applications by using heuristics.
- Monitoring of the resources under its control in terms of use, and report the corresponding meta-broker instance for global decision-making.
- Accountability of the submitted jobs in terms of different states e.g., submitted, pending, executed, and failed.

3.6.3. Resource Layer (RL)

The infrastructure layer is the low-level layer, which comprises of computational resources (CPUs), storage resources (RAM, hard disks etc.), virtualization of hardware and network resources (link capacity), and other resources for services (applications) that support cloud system activities. The resources on this layer are pre-configured and maintained by service providers. The proposed system depicted in Figure 2 is a decentralized Intercloud scheduling architecture, which satisfies performance criteria including scalability, reliability, and availability while handling the dynamicity.

3.7. Application Model and Dynamic Allocation

To simulate the workload in federated cloud domains, PSAs are defined as a set of n independent jobs $\{J_i\}$ $i = 1 \dots n$. The processing requirement of these jobs is considered to be the length in Millions of Instructions (MIs), the size of job input and output data in bytes, along with various other execution related parameters. Random length of jobs is used to model heterogeneous tasks like in real-time parameter-sweep applications. Task-framing applications are comprised of different sets of jobs, e.g., (200, 500, 1000) with different MIs.

Since the proposed system comprises various distinct modules they execute different layers to keep the decoupled settings of the overall system. Modular structure as depicted in Figure 6 simplifies the complexity of the system and the adaptive control of the dynamicity for the management of resources. End users submit applications with specific parameters to the associated peer (meta-broker) for execution at a particular time interval. Each meta-brokering instance communicates with the corresponding cloud through the local-broker to constantly maintain the latest information of resource states with attributes, e.g., static and dynamic. Acquiring the latest resource information is essential for managing unpredictable cloud systems. The application cycle in the proposed system is depicted in Figure 7, as jobs are traversed through different phases to get the execution completed.

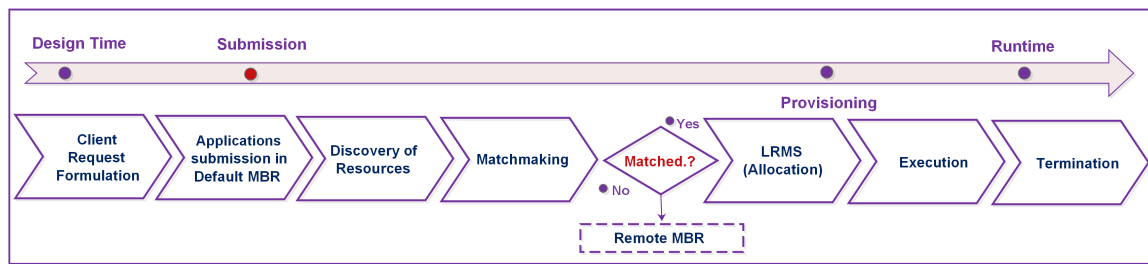


Figure 7. Applications in different phases.

In the proposed dynamic resource allocation policy at datacenter level, the load on all virtual resources (VMs) is taken into account and the completion time of current load is identified. After this, the completion time of the currently received job is calculated on each resource, and this time is incorporated into the existing load (completion time) of each resource. In this way, the least load's completion time of virtual resource is obtained from this calculation, and the job is allocated to that particular least-loaded resource (VM). Since the proposed system has been implemented in the simulation environment for the service provision of PSAs with varying workload, this dynamic allocation policy for such applications is most suitable in terms of heterogeneous resources at the datacenter. Figure 8 depicts the algorithmic flow of the dynamic resource allocation scheme.

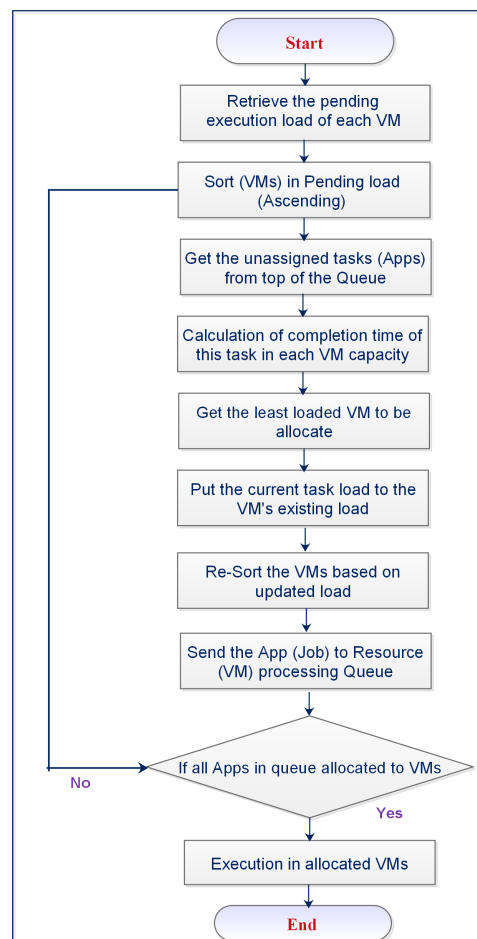


Figure 8. Dynamic scheduling at datacenter level.

4. Problem Formulation

The facilitation of scalable provision of application services (parametric sweep) under varying workload requires effective brokering policies in the cloud federation to find optimal service locations.

Therefore, developing the techniques of workload distribution and dynamic resource allocation for application services is imperative for attaining better QoS. The proposed model consists of mathematical sets, which indicate various elements as follows:

The set CSP indicates the CSPs, which are participating in the federated system as

$$CSP = \{CSP_i, CSP_i + 1, \dots, CSP_n\} \quad (1)$$

$ICMB$ is set of intercloud meta-brokers (ICMB) associated with each service provider as

$$ICMB = \{ICMB_i, ICMB_i + 1, \dots, ICMB_n\} \quad (2)$$

Each service provider is available with a set of Hosts and VMs with different Mips for end-user service execution as

$$VMs = \{VM_j, VM_j + 1, \dots, VM_n\} \quad (3)$$

Similarly, a set of end-user applications is described as

$$Apps = \{App_i, App_i + 1, \dots, APP_n\} \quad (4)$$

The processing capacity of any individual VM is calculated in the following equation, where PE is the processing element and $mipsPE$ is Millions of instructions per second of PE .

$$capacityVM_k = \sum_{j=1}^m PE_j * mipsPE_j \quad \text{where } k \in VMs = \{1, 2, 3, \dots, n\} \quad (5)$$

The processing capacity of single CSP is the total capacity of overall virtual machines (VMs) deployed in particular CSPs calculated as

$$CapacityCSP_i = \sum_{j=1}^m capacityVM_j \quad \text{where } i \in CSP = \{1, 2, 3, \dots, n\} \quad (6)$$

Basically, the capacity of a single cloud is the threshold value for application-forwarding decisions at the ICMB level. Therefore, total computational capacity of all interconnected federated CSPs is calculated as

$$capacityTotal_{Fed} = \sum_{i=1}^n CapacityCSP_i \quad (7)$$

Variable workload information on VM_j is calculated by determining the queue length of VM scheduler at time interval t by using the following equation.

$$LVM_{i,t} = Q * TL_{i(t)} / S(VM_{i,t}) \quad (8)$$

where $Q = \{1, 2, 3, \dots, n\}$ are user application instances in the VM_i queue. The workload on the virtual machine at time interval t is calculated by taking the ratio of task length on particular VM_i processing power (Mips) or service rate. $S(VM_{i,t})$ defines the service rate of VM_i at time interval t and the scheduling algorithm of local-broker achieves the fitness of VM_i using this equation. The workload on all virtual machines (VMs) is calculated as

$$LVMs_{total} = \sum_{j=1}^m LVM_{j,t} \quad (9)$$

$LVMs_{total}$ is the threshold value for the decision of the forwarding applications and indicates the overall load on particular CSP at any given time. The processing time of applications on the resource VM_j is obtained as

$$PT_{VMj} = \sum_{i=1}^n E_{ij} * TL_i / PE_j * mipsPE_j, \quad (10)$$

where TL_i is task length(MIs) and decision variable $E_{ij} = 1$ if the *application* job_i is allocated to VM_j and 0 otherwise. The objectives of interconnected cloud scheduling are as follows:

- In overwhelming conditions of a particular federated site, to maximize the routing (forwarding) the user applications towards remote CSP to facilitate user workload scalability and improve the confidence of enterprises on cloud technologies.
- Minimize the processing time of user applications on the CSP datacenter.

The LP model to maximize the application routing: Let $Route_{ij(t)}$ be the forwarding function of App_i towards $ICMB_j$ to scale the workload at time interval t , when particular sites get overwhelming. $Route_{ij(t)}$ functions correspond to Algorithm 1. Each $ICMB_j$ instance is associated with each CSP and the LP model is defined as:

$$\text{maximize } M = \sum_{i=1}^m \sum_{j=1}^n Y_{ij} Route_{ij(t)} \quad (11)$$

where $j \in ICMB = \{1, 2, 3, \dots, n\}$ and $i \in Apps = \{1, 2, 3, \dots, m\}$, while decision variable $Y_{ij} = 1$ if the App_i is forwarded to remote $ICMB_j$ and 0 otherwise, under the following constraints:

$$\sum_{i=1}^m Y_{ij} LVMs_{total} \geq Cloud_{Threshold} \forall i \in Apps \quad (12)$$

$$\sum_{i=1}^n Y_{ij} = 1 \forall i \in Apps \quad (13)$$

The constraint set (Equation (12)) describes that applications will be forwarded in case exceeding the load from a certain threshold value. The constraint set (Equation (13)) specifies that applications will be routed towards only one site. Application submissions to meta-broker $ICMB_i$ interface are under the conditions as follows:

$$SubmitApps_i = \begin{cases} ICMBi_{remote} & \text{if } (LVMs_{total} \geq CapacityCSP_i) \forall i \in APPs = \{1, 2, 3, \dots, n\} \\ ICMBi_{default} & \text{otherwise} \end{cases} \quad (14)$$

The LP model for minimizing the processing time is as: Let PT_{ij} be the processing time of allocating application i to VM j and defined as:

$$X_{ij} = \begin{cases} 1 & \text{if } (Apps \text{ } p \text{ is assigned}) \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

$$\text{minimize } Z = \sum_{i=1}^n \sum_{j=1}^m PT_{ij(t)} X_{ij} \quad (16)$$

Least-loaded resource (VM) will be allocated for application by dynamic scheduler to minimize the processing time (PT) at time interval t , Which is subjected to the following:

$$\sum_{i=1}^n X_{ij} = 1 \forall j \in VMs = \{1, 2, 3, \dots, m\} \quad (17)$$

$$\sum_{i=1}^n X_{ij} \leq PE_j \forall j \in VMs = \{1, 2, 3, \dots, m\} \quad (18)$$

where decision variable $X_{ij}=1$ if task i is assigned for all VMs and the required processing element for execution must be less or equal to the available in VMs.

4.1. Implementation Strategy

Experimentation on the real environment to evaluate the proposed system is extremely costly, complex, and time-consuming. Therefore, an alternative solution to conduct such experiments is a simulation medium. The CloudSim toolkit (simulation library) developed by Cloud Laboratory [48] is extendable, and highly customized APIs can be harnesses to test the diversified cloud systems. CloudSim is discrete-event simulation library and benchmark to model the proposed system with different parameters in a repeatable manner. Therefore, to carry out the simulation-driven implementation of the proposed approach, various classes of CloudSim have been extended to model and realize the desired behavior of system entities. Mainstream classes that are indispensable for any scenario are mentioned below. Simulation flow using these classes is depicted in Figure 9. These entities communicate by sending messages of specific information from source to destination.

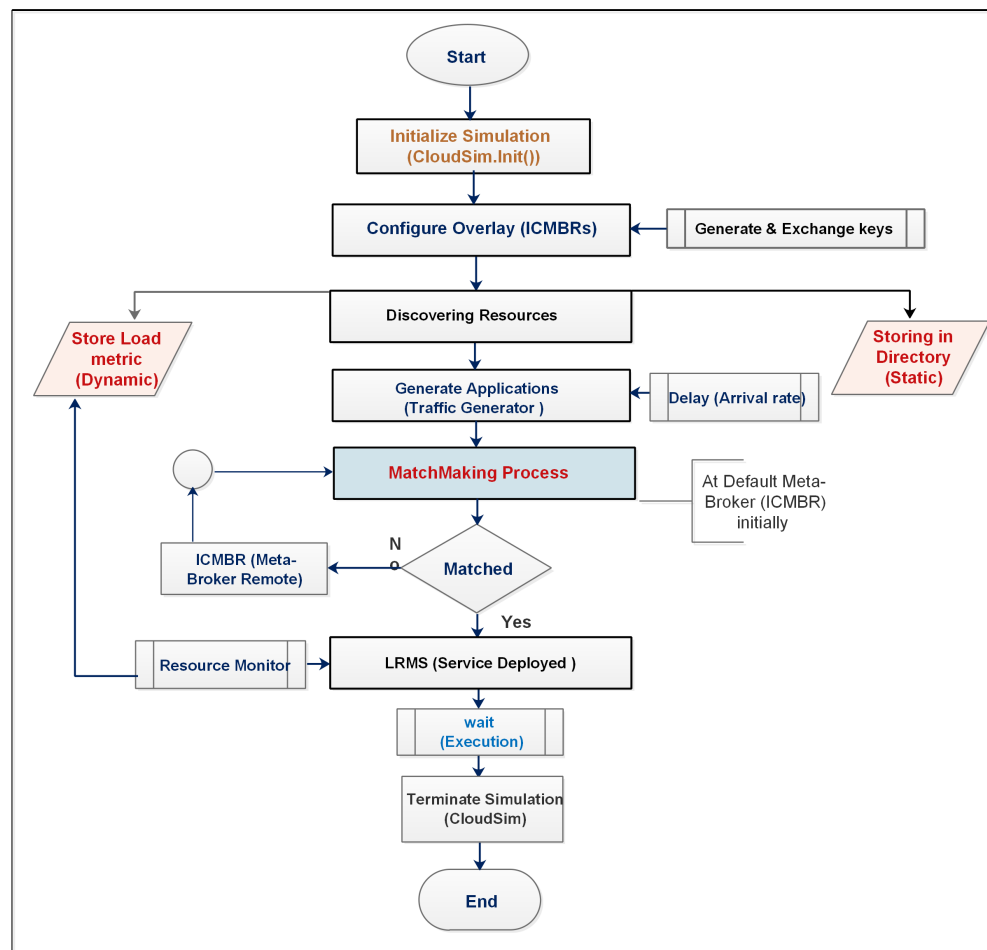


Figure 9. Simulation Flow of Proposed System.

- + **Datacenter:** Datacenter class used to model the IaaS provider with different configuration of server machines.
- + **DatacenterBroker:** Function of this class is to mediate the operation between end users and provider.
- + **SimEntity:** To develop the new entity from scratch, this class must be extended.

- + **Host:** It represents the physical machine with different capacity of resources, e.g., processor, RAM, bandwidth.
- + **VM:** To provide the virtualization on host node, and applications are sand-boxed in VMs.
- + **Cloudlet:** End-user applications with different execution instructions modeled through this class.
- + **SimEvent:** These objects represent the messages/events with specific information that simulation entities send to each other, and takes appropriate action accordingly.
- + **CloudInformationService:** Registration of resources, indexing, and discovery are facilitated through this CIS class.

4.2. Simulation Results Performance

In this section, we describe the expected performance of the proposed model. Experiments are carried out using CloudSim simulation libraries to validate the model with different number of user applications to check the scalability and reliability of the system. Figure 10 depicts the deployment rate of user applications on joint clouds in federated schema. Applications with varying MIs within the random range from 100,000 to 1,270,000 are well scaled on participating clouds during execution.

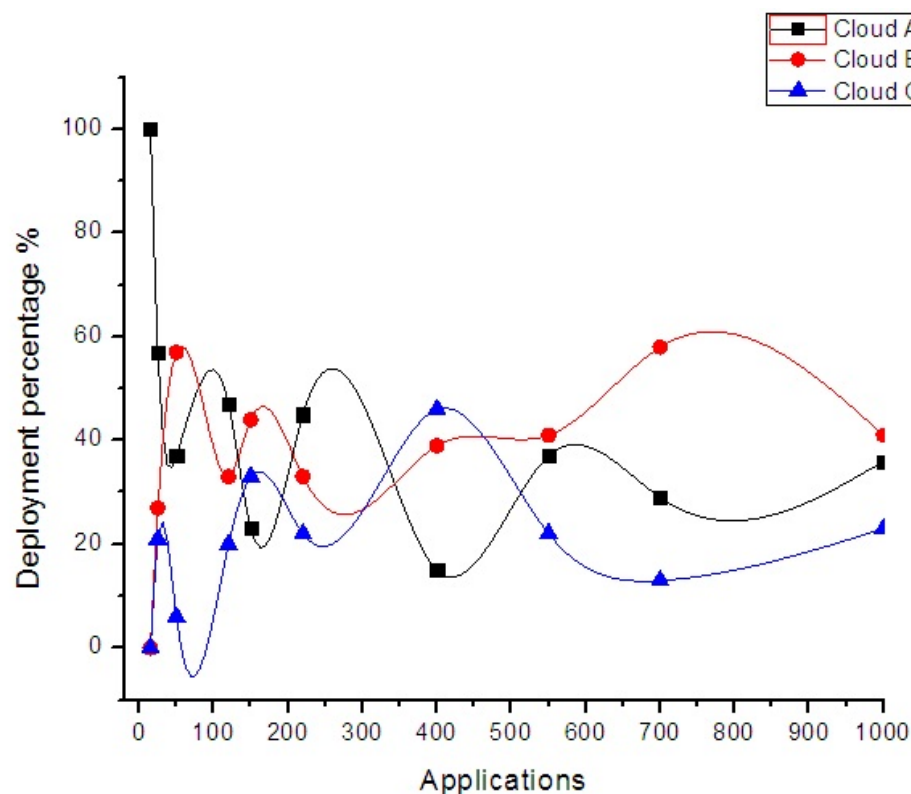


Figure 10. Deployment rate of user applications.

Experiments to analyze the variation in scaling process with different numbers of clouds are depicted in Figure 11. As the number of clouds increases, the scaling process will be flexible and, consequently, better performance will be achieved.

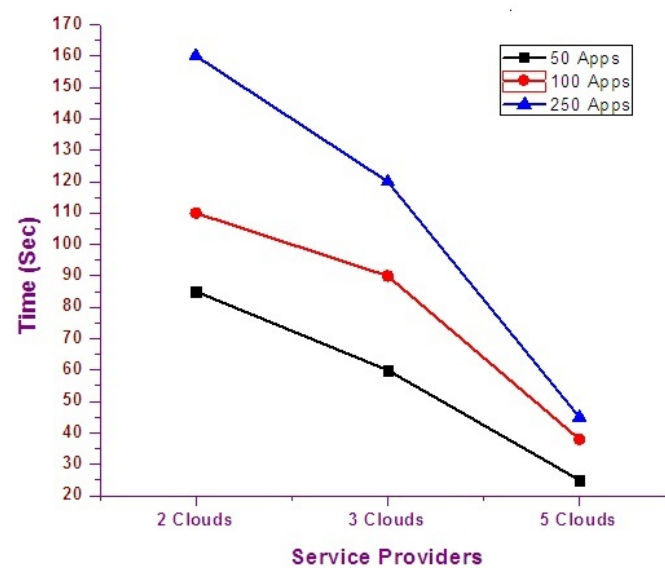


Figure 11. Variation of (PT) in Different Environment.

Experimental analysis of the application burst is depicted in Figure 12, which indicates that after reaching a certain range of particular cloud Mips (e.g., after consumption of available capacity), user applications are being burst into other participated sites. This factor is achieved based on data structure information. This information is obtained through underlying monitoring modules and facilitates dynamic brokerage and scaling of the workload accordingly. Figure 13 depicts the waiting time of the user applications in both the scaling and non-scaling mode. In the scaling mode, the waiting time of the application is significantly decreased because of adequate availability of resource in the form of joint clouds.

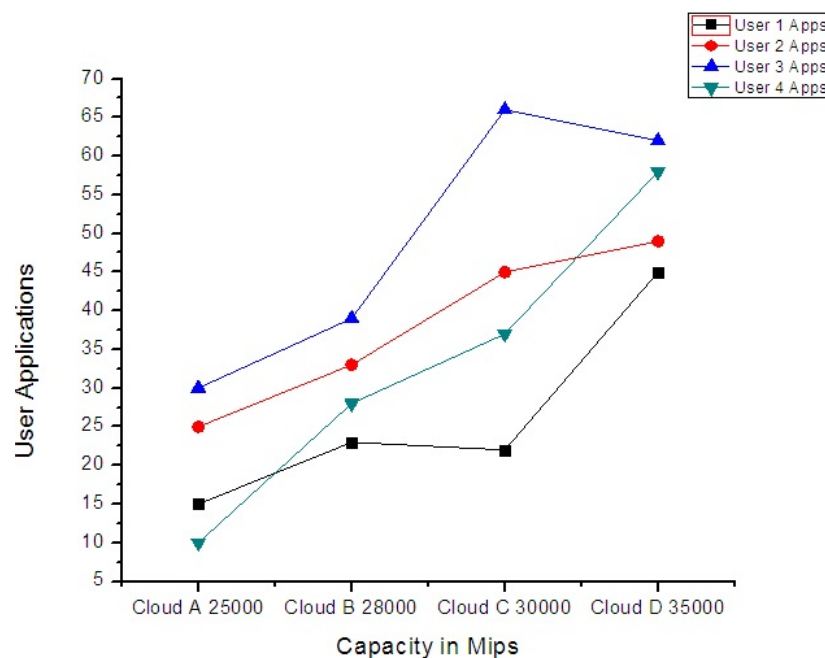


Figure 12. Applications bursting into various clouds.

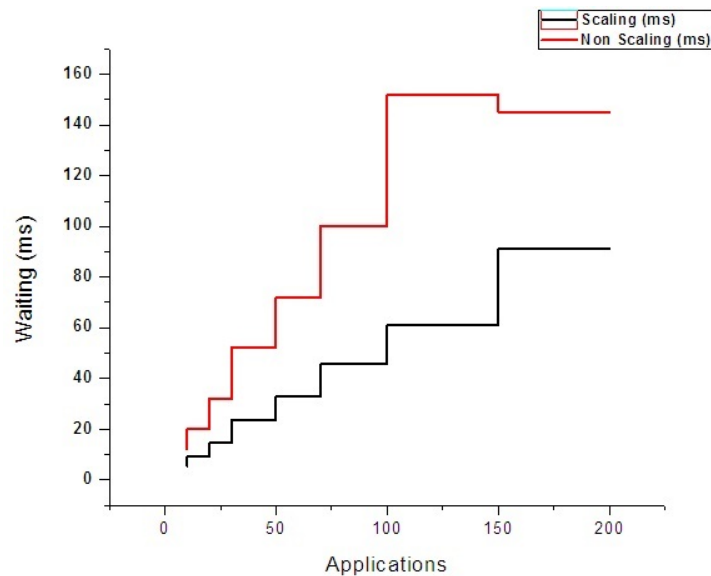


Figure 13. Applications waiting time.

The analysis of user satisfaction in terms of application requirement (e.g., Mips rate, OS) is depicted in Figure 14. As the number of clouds increases, the satisfaction level also increases due to high scalability of applications among more clouds. The percentage of satisfaction indicates the degree to which much resources have been met for application execution with 2, 3, and 5 datacenters.

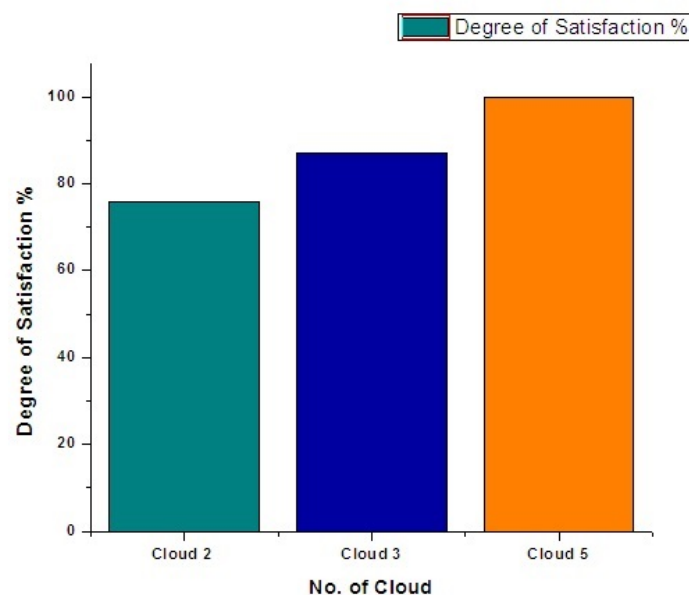


Figure 14. Degree of user satisfaction level.

Figure 15 demonstrates the performance results of another experiment of 500 user applications with varying workloads within the range of 150,000 to 170,000 MIs. The addition of two more clouds significantly decreases the processing time (PT) of user application. When a particular datacenter reaches a certain threshold capacity, the incoming workload is routed to the next one and the processing time tends to stabilize. Figure 16 depicts the use of each cloud resource under threshold limit and the exceeding workload is forwarded to other clouds in the federated schema. The criterion to keep use under a certain threshold helps to avoid the overwhelming condition of any participating cloud.

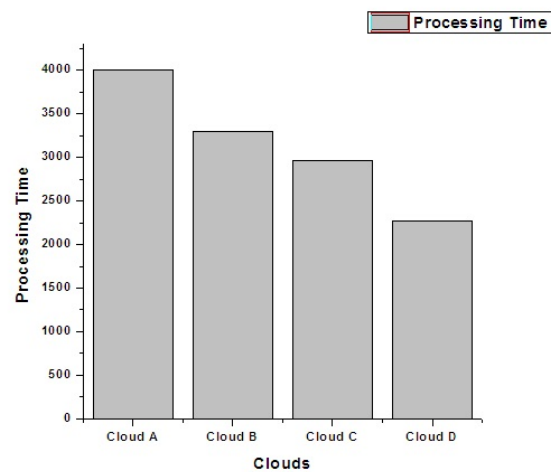


Figure 15. Total processing time.

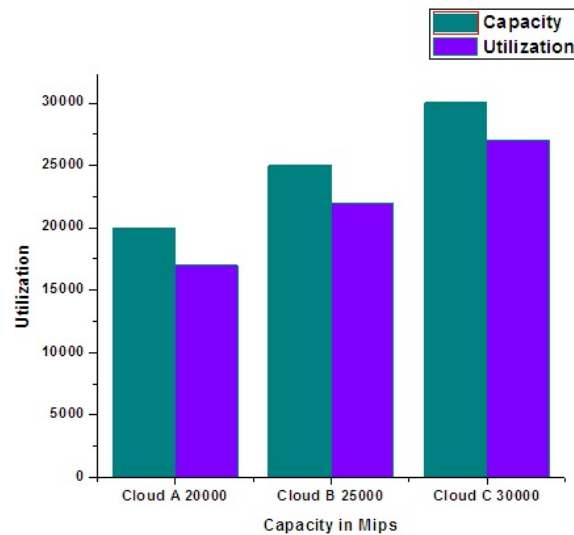


Figure 16. Use of cloud capacity.

Figure 17 depict the average execution result of user application on local datacenter level.

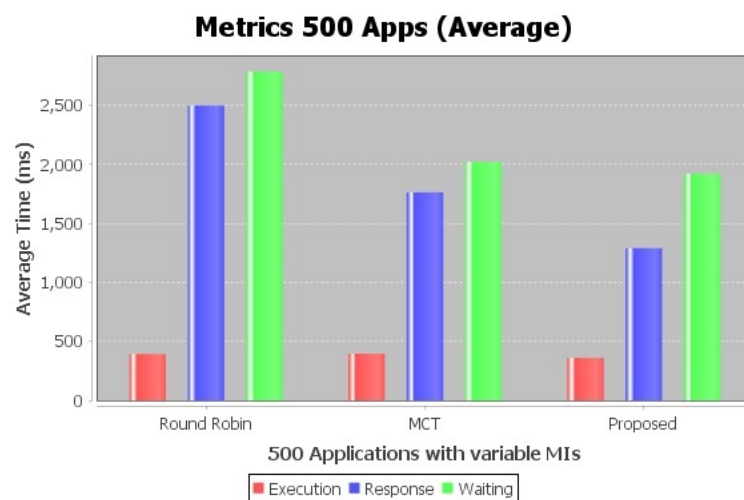


Figure 17. Application (500) performance at local (DC) level.

Figure 18 depicts the performance with 5 clouds, which indicates the makespan of different applications and analyzes the efficiency of the federation process with increasing clouds.

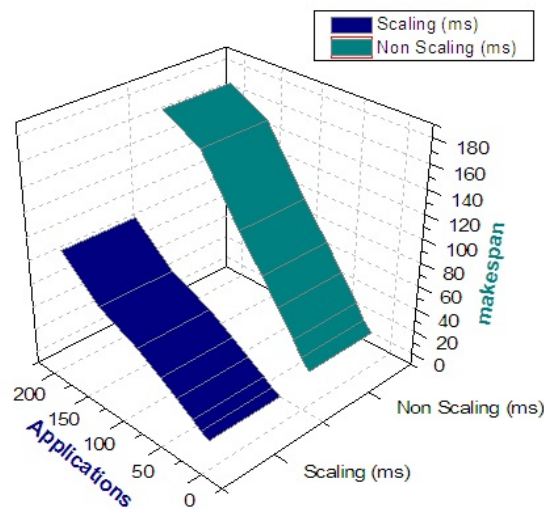


Figure 18. Makespan with different applications.

In Figure 19, the application acceptance rate is analyzed as resources are being increased. Acceptance rate is high due to availability of many resources. There are many chances of meeting the required application attributes due to the large availability of a large amount of resources.

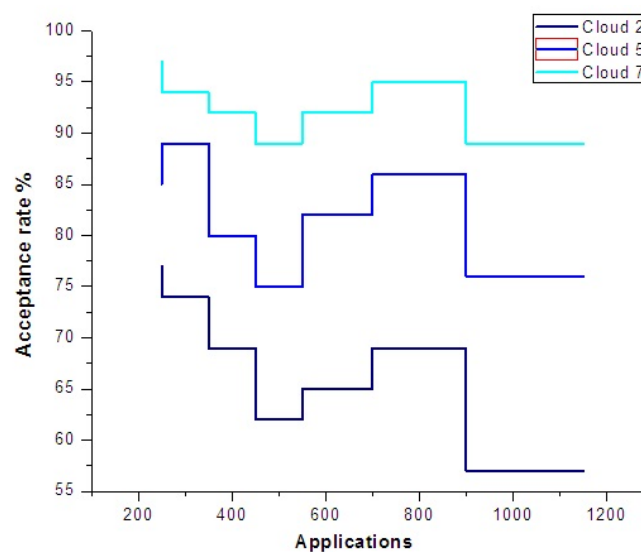


Figure 19. Application acceptance rate.

Figure 20 depicts the response time of user applications on different clouds by using different algorithms with comparison of the proposed approach on different set of clouds.

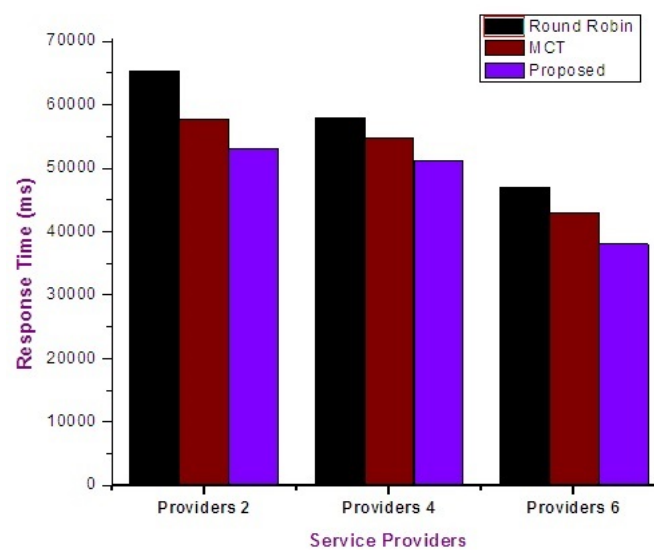


Figure 20. Response time.

5. Concluding Remarks

Interconnecting different clouds is indispensable to overcome the overwhelming condition of standalone service providers. Scheduling design and brokering factors play significant roles in the development of robust and scalable systems. These systems are essential to achieve optimal resource management in highly dynamic and distributed environments. In this paper, decentralized meta-brokering-driven overlay architecture is proposed for Intercloud scheduling services. Meta-brokering peers establish links at application level and communicate in a P2P manner. This approach provides more scalability compared to its centralized counterparts, and maintains the real-time statistics of peers in overlay to facilitate dynamic brokerage. The threshold criterion of dynamic attributes of resources has adapted to avoid the overwhelming state of a particular domain. Routing of user applications is accomplished based on data structure maintained by each peer (meta-broker). These peers exchange information to control the dynamicity of a system and enable the service provision from optimal locations. Since research in the cloud federation is still in its infancy, standard interfaces, uniform architectural design, and seamless communication protocols are required to overcome issues, e.g., application portability, interoperability, and heterogeneity factors, among CSPs. Similarly, further research is required to develop optimal brokering techniques for enterprise applications (e.g., web-centric business and interactivity) to improve QoS metrics. Regarding validation and evaluation, the proposed design was implemented by extending the CloudSim 3.0.3 simulation library (APIs), and obtained satisfactory results.

Author Contributions: Design's concept, development, simulation and testing; and manuscript editing and compilation by S.L.; Supervision of the whole research activity by S.M.G.; R.L.A and M.L. facilitated the technical tools, Lab experimentation, debugging and tweaking the write-up. Overall organization and refinement of article by K.-M.K.

Funding: This research was funded by Basic Sciences Research Program through the National Research Foundation of Korea (NRF) Funded by the Ministry of Education (2017030223).

Acknowledgments: Authors are thankful to the PMAS-Arid University for facilitating this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ouardi, A.; Sekkaki, A.; Mammass, D. Towards an inter-Cloud architecture in healthcare system. In Proceedings of the 2017 International Symposium on Networks, Computers and Communications (ISNCC), Marrakech, Morocco, 16–18 May 2017; pp. 1–6.

2. Kertész, A.; Kecskemeti, G.; Brandic, I. An interoperable and self-adaptive approach for SLA-based service virtualization in heterogeneous Cloud environments. *Future Gener. Comput. Syst.* **2014**, *32*, 54–68. [\[CrossRef\]](#)
3. Salama, M.; Shawish, A. A QoS-oriented inter-cloud federation framework. In Proceedings of the 2014 IEEE 38th Annual Computer Software and Applications Conference (COMPSAC), Vasteras, Sweden, 21–25 July 2014; pp. 642–643.
4. Assis, M.R.; Bittencourt, L.F. A survey on cloud federation architectures: Identifying functional and non-functional properties. *J. Netw. Comput. Appl.* **2016**, *72*, 51–71. [\[CrossRef\]](#)
5. Liaqat, M.; Chang, V.; Gani, A.; Ab Hamid, S.H.; Toseef, M.; Shoaib, U.; Ali, R.L. Federated cloud resource management: Review and discussion. *J. Netw. Comput. Appl.* **2017**, *77*, 87–105. [\[CrossRef\]](#)
6. Buyya, R.; Ranjan, R.; Calheiros, R.N. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing, Busan, Korea, 21–23 May 2010; Springer: Berlin, Germany, 2010; pp. 13–31.
7. Esposito, C.; Ficco, M.; Palmieri, F.; Castiglione, A. Interconnecting federated clouds by using publish-subscribe service. *Clust. Comput.* **2013**, *16*, 887–903. [\[CrossRef\]](#)
8. Erdil, D.C. Autonomic cloud resource sharing for intercloud federations. *Future Gener. Comput. Syst.* **2013**, *29*, 1700–1708. [\[CrossRef\]](#)
9. Lloret, J.; Garcia, M.; Tomas, J.; Rodrigues, J.J. Architecture and protocol for intercloud communication. *Inf. Sci.* **2014**, *258*, 434–451. [\[CrossRef\]](#)
10. Kelaidonis, D.; Rouskas, A.; Stavroulaki, V.; Demestichas, P.; Vlascheas, P. A federated edge Cloud-IoT architecture. In Proceedings of the 2016 European Conference on Networks and Communications (EuCNC), Athens, Greece, 27–30 June 2016; pp. 230–234.
11. Hoenisch, P.; Hochreiner, C.; Schuller, D.; Schulte, S.; Mendling, J.; Dustdar, S. Cost-efficient scheduling of elastic processes in hybrid clouds. In Proceedings of the 2015 IEEE 8th International Conference on Cloud Computing, New York, NY, USA, 27 June–2 July 2015; pp. 17–24.
12. Petcu, D. Consuming resources and services from multiple clouds. *J. Grid Comput.* **2014**, *12*, 321–345. [\[CrossRef\]](#)
13. Fazio, M.; Celesti, A.; Villari, M.; Puliafito, A. How to enhance cloud architectures to enable cross-federation: Towards interoperable storage providers. In Proceedings of the 2015 IEEE International Conference on Cloud Engineering, Tempe, AZ, USA, 9–13 March 2015; pp. 480–486.
14. Bittencourt, L.F.; Madeira, E.R.; Da Fonseca, N.L. Scheduling in hybrid clouds. *IEEE Commun. Mag.* **2012**, *50*, 42–47. [\[CrossRef\]](#)
15. Calcavecchia, N.M.; Celesti, A.; Di Nitto, E. Understanding decentralized and dynamic brokerage in federated cloud environments. In *Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice: Theory and Practice*; IGI Global: Hershey, PA, USA, 2012; p. 36.
16. Khanna, P.; Jain, S. Distributed cloud federation brokerage: A live analysis. In Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, London, UK, 8–11 December 2014; pp. 738–743.
17. Grozev, N.; Buyya, R. Inter-Cloud architectures and application brokering: Taxonomy and survey. *Softw. Pract. Exp.* **2014**, *44*, 369–390. [\[CrossRef\]](#)
18. Rani, B.K.; Rani, B.P.; Babu, A.V. Cloud computing and inter-clouds—types, topologies and research issues. *Procedia Comput. Sci.* **2015**, *50*, 24–29. [\[CrossRef\]](#)
19. Kogias, D.G.; Xevgenis, M.G.; Patrikakis, C.Z. Cloud federation and the evolution of cloud computing. *Computer* **2016**, *49*, 96–99. [\[CrossRef\]](#)
20. Slawik, M.; Zilci, B.I.; Demchenko, Y.; Baranda, J.I.A.; Branchat, R.; Loomis, C.; Lodygensky, O.; Blanchet, C. CYCLONE unified deployment and management of federated, multi-cloud applications. In Proceedings of the 8th International Conference on Utility and Cloud Computing, Limassol, Cyprus, 7–10 December 2015; pp. 453–457.
21. Aazam, M.; Huh, E.N. Framework of resource management for intercloud computing. *Math. Probl. Eng.* **2014**, *2014*, 108286. [\[CrossRef\]](#)
22. Lee, C.A. Cloud federation management and beyond: Requirements, relevant standards, and gaps. *IEEE Cloud Comput.* **2016**, *3*, 42–49. [\[CrossRef\]](#)

23. Kertész, A.; Dombi, J.D. Adaptive scheduling solution for grid meta-brokering. *Acta Cybern.* **2009**, *19*, 105–123. [\[CrossRef\]](#)
24. Sotiriadis, S.; Bessis, N.; Antonopoulos, N. Decentralized meta-brokers for inter-cloud: Modeling brokering coordinators for interoperable resource management. In Proceedings of the 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, Sichuan, China, 29–31 May 2012; pp. 2462–2468.
25. García, Á.L.; del Castillo, E.F.; Fernández, P.O. Standards for enabling heterogeneous IaaS cloud federations. *Comput. Stand. Interfaces* **2016**, *47*, 19–23. [\[CrossRef\]](#)
26. Ranjan, R. The cloud interoperability challenge. *IEEE Cloud Comput.* **2014**, *1*, 20–24. [\[CrossRef\]](#)
27. Arunkumar, G.; Venkataraman, N. A novel approach to address interoperability concern in cloud computing. *Procedia Comput. Sci.* **2015**, *50*, 554–559. [\[CrossRef\]](#)
28. Di Costanzo, A.; De Assuncao, M.D.; Buyya, R. Harnessing cloud technologies for a virtualized distributed computing infrastructure. *IEEE Internet Comput.* **2009**, *13*, 24–33. [\[CrossRef\]](#)
29. Calheiros, R.N.; Toosi, A.N.; Vecchiola, C.; Buyya, R. A coordinator for scaling elastic applications across multiple clouds. *Future Gener. Comput. Syst.* **2012**, *28*, 1350–1362. [\[CrossRef\]](#)
30. Sotiriadis, S.; Bessis, N.; Anjum, A.; Buyya, R. An inter-cloud meta-scheduling (icms) simulation framework: Architecture and evaluation. *IEEE Trans. Serv. Comput.* **2018**, *11*, 5–19. [\[CrossRef\]](#)
31. Amin, M.B.; Khan, W.A.; Awan, A.A.; Lee, S. Intercloud message exchange middleware. In Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, Kuala Lumpur, Malaysia, 20–22 February 2012; p. 79.
32. Nelson, V.; Uma, V. Semantic based resource provisioning and scheduling in inter-cloud environment. In Proceedings of the 2012 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, India, 19–21 April 2012; pp. 250–254.
33. Manno, G.; Smari, W.W.; Spalazzi, L. Fcfa: A semantic-based federated cloud framework architecture. In Proceedings of the 2012 International Conference on High Performance Computing & Simulation (HPCS), Madrid, Spain, 2–6 July 2012; pp. 42–52.
34. Jrad, F.; Tao, J.; Streit, A. SLA based Service Brokering in Intercloud Environments. In Proceedings of the 2nd International Conference on Cloud Computing and Services Science, Porto, Portugal, 18–21 April 2012; pp. 76–81.
35. Khanna, P.; Jain, S.; Babu, B. Distributed cloud brokerage: solution to real world service provisioning problems. *ARPJ J. Eng. Appl. Sci.* **2015**, *10*, 2011–2016.
36. De Assunção, M.D.; Buyya, R. Performance analysis of allocation policies for interGrid resource provisioning. *Inf. Softw. Technol.* **2009**, *51*, 42–55. [\[CrossRef\]](#)
37. Iosup, A.; Tannenbaum, T.; Farrellee, M.; Epema, D.; Livny, M. Inter-operating grids through delegated matchmaking. *Sci. Program.* **2008**, *16*, 233–253. [\[CrossRef\]](#)
38. Takabi, H.; Joshi, J.B. Semantic-based policy management for cloud computing environments. *Int. J. Cloud Comput.* **2012**, *1*, 119–144. [\[CrossRef\]](#)
39. Brabara, H.; Mtibaa, A.; Sliman, L.; Gaaloul, W.; Gargouri, F. Semantic web technologies in cloud computing: a systematic literature review. In Proceedings of the 2016 IEEE International Conference on Services Computing (SCC), San Francisco, CA, USA, 27 June–2 July 2016; pp. 744–751.
40. Nakamura, L.H.; Estrella, J.C.; Santana, R.H.; Santana, M.J.; Reiff-Marganiec, S. A semantic approach for efficient and customized management of IaaS resources. In Proceedings of the 10th International Conference on Network and Service Management (CNSM) and Workshop, Rio de Janeiro, Brazil, 17–21 November 2014; pp. 360–363.
41. Huedo, E.; Montero, R.S.; Moreno, R.; Llorente, I.M.; Levin, A.; Massonet, P. Interoperable federated cloud networking. *IEEE Internet Comput.* **2017**, *21*, 54–59. [\[CrossRef\]](#)
42. Nodehi, T.; Jardim-Goncalves, R.; Zutshi, A.; Grilo, A. ICIF: an inter-cloud interoperability framework for computing resource cloud providers in factories of the future. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 147–157. [\[CrossRef\]](#)
43. Levin, A.; Barabash, K.; Ben-Itzhak, Y.; Guenender, S.; Schour, L. Networking architecture for seamless cloud interoperability. In Proceedings of the 2015 IEEE 8th International Conference on Cloud Computing, New York, NY, USA, 27 June–2 July 2015; pp. 1021–1024.

44. Iamnitchi, A.; Foster, I.; Nurmi, D. A peer-to-peer approach to resource discovery in grid environments. In Proceedings of the IEEE High Performance Distributed Computing, Edinburgh, UK, 24–26 July 2002; Volume 140, pp. 122–222.
45. Iamnitchi, A.; Foster, I. A peer-to-peer approach to resource location in grid environments. In *Grid Resource Management*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 413–429.
46. Iamnitchi, A. Resource Discovery in Large-Scale Distributed Environments. Ph.D. Thesis Proposal, University of Chicago, Chicago, IL, USA, May 2002.
47. Li, W.; Xu, Z.; Dong, F.; Zhang, J. Grid resource discovery based on a routing-transferring model. In *Grid Computing GRID 2002*; Springer: Berlin/Heidelberg, Germany, 2002; Volume 78, pp. 145–156.
48. Buyya, R.; Ranjan, R.; Calheiros, R.N. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In Proceedings of the High Performance Computing & Simulation, Leipzig, Germany, 21–24 June 2009; pp. 1–11.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).