

Global Manager – A Service Broker In An Integrated Cloud Computing, Edge Computing & IoT Environment

Kailash Selvaraj^{1*}, and Saswati Mukherjee²

¹Centre for Development of Advanced Computing (C-DAC), Chennai, India.
[e-mail: kailashs@cdac.in]

²College of Engineering, Guindy, Anna University, Chennai, India.
[e-mail: msaswati@auist.net]

*Corresponding author: Kailash Selvaraj

Received May 26, 2021; accepted January 6, 2022; published June 30, 2022

Abstract

The emergence of technologies like Big data analytics, Industrial Internet of Things, Internet of Things, and applicability of these technologies in various domains leads to increased demand in the underlying execution environment. The demand may be for compute, storage, and network resources. These demands cannot be effectively catered by the conventional cloud environment, which requires an integrated environment. The task of finding an appropriate service provider is tedious for a service consumer as the number of service providers drastically increases and the services provided are heterogeneous in the specification. A service broker is essential to find the service provider for varying service consumer requests. Also, the service broker should be smart enough to make the service providers best fit for consumer requests, ensuring that both service consumer and provider are mutually beneficial. A service broker in an integrated environment named Global Manager is proposed in the paper, which can find an appropriate service provider for every varying service consumer request. The proposed Global Manager is capable of identification of parameters for service negotiation with the service providers thereby making the providers the best fit to the maximum possible extent for every consumer request. The paper describes the architecture of the proposed Global Manager, workflow through the proposed algorithms followed by the pilot implementation with sample datasets retrieved from literature and synthetic data. The experimental results are presented with a few of the future work to be carried out to make the Manager more sustainable and serviceable.

Keywords: Integrated cloud edge environment, Service broker, Service negotiation

1. Introduction

Emerging technologies like the Internet of Things (IoT), Industrial Internet of Things (IIoT), Big Data Analytics, Artificial Intelligence, Machine Learning, Inter Cyber-Physical Systems (ICPS), Industry 4.0 utilize the Cloud environment for their computational and storage needs. Such use cases for the cloud possess specific requirements which are heterogeneous in nature, including less latency, need for processing environment near the source of data, performance, security, fewer communication overheads, and various other factors that provide less opportunity to utilize cloud services directly. Conventional cloud services like IaaS, PaaS, and SaaS may not be able to effectively cater to these heterogeneous requirements. This scenario is explained by the introduction of the Cross-Cloud Computing approach [44] to address the interoperability, migration, and deployment strategies. On the other side, a tremendous increase in the number of cloud service providers with heterogeneity in their service offerings make the selection of appropriate cloud service challenging for the consumers [39].

Certain computational and storage requirements can be catered by incorporating or integrating Edge computing / Fog computing with a conventional cloud. Recent research strongly focuses on the special needs of IoT applications that can be catered to by an integrated environment comprising Edge, Fog, and Cloud infrastructures [31]. Another variety of integration is presented in the form of federated cloud, inter-cloud, hybrid cloud, multi-cloud and cross-cloud where the integration comprises only cloud environments [39]. With resource constraints in the form of computing, storage, network, Edge / Fog nodes require support of the Cloud for successful execution of consumer requests with better quality of service thereby bringing in the need for an integrated environment. The expansion of service consumption transforms the underlying computation environment into an integrated environment comprising Cloud, Edge / Fog, and Gateway. The wide increased consumption of services with specific domain requirements provide an opportunity for expansion of the underlying execution of cloud environment to other forms such as Edge / Fog environment, IoT Gateway for communication and control, Software-Defined Networking (SDN) or Software-Defined Wide Area Network (SD-WAN) framework for automation of network management, orchestration, performance, and security.

As the variety of services provided increases in the integrated environment, the category of service specification by the providers also expands as each provider will have their way of specifying the services. Due to heterogeneity and dynamic changes in cloud services, there is a lack of systematic approach for the comparison with other similar services [32], [24]. This may be attributed to the absence of a standardized mechanism of service provider specification approach [43]. This is applicable for Edge nodes as well. On the other hand, the request specifications by consumers vary with their specific parameters. In the integrated environment, the cloud environment, edge nodes, IoT Gateway or users play the role of service consumers. The service request by all these entities varies to a larger extent. Thus, the service provider selection process is becoming complicated, bringing in the need for a comprehensive selection methodology [30]. The challenges for consumers are in selecting the best service as it is time-consuming to analyze and choose the service provider [1], [21]. Hence certain level of automation is required to analyze the heterogeneous request, service offering specification and selecting the appropriate service provider corresponding to the consumer request.

Finding an appropriate service provider corresponding to consumer requests is the first challenge in effective utilization of the integrated environment that can be solved by a Cloud

broker [1]. The serviceability of cloud brokers is to be extended to the integrated environment comprising edge / fog nodes, gateway devices, hence termed as service broker in general. Guaranteeing effective trading to both seller and buyer is an open problem yet to be addressed [20]. The lack of standardized comparison among service offerings leads to challenging decision making [38]. The coordination of an integrated environment with diversity and heterogeneity in operation is a challenging task [31]. To address these concerns, the service broker is essential. The Service broker generally acts as an entity between the service consumer and service provider to perform various functions and the most critical function of matching the service request with the provider [1]. There are various cloud brokers as revealed in the literature [1], [14], [15], [17], [30], [32], [38], [41], [42]. These brokers are generally constrained to the cloud environment and lack focus on the specified integrated environment. Hence, a service broker is essential in an integrated Cloud, Edge / Fog and IoT environment. Apart from selecting the service provider, the service broker must be smart enough to negotiate with the provider to fit the service consumer specifications best, when there is a deficit in service offered by the provider for consumer requirements. Generally, Cloud service broker functionality includes decision support, monitoring of resources and services, enforcement of policy, SLA compliance and negotiation, workload deployment, migration, API abstraction, interoperability and portability orchestration, life cycle management, evaluation, etc. [39].

This paper introduces a smart service broker termed as Global Manager (GM), serving the integrated environment (Cloud, Edge / Fog, Gateway, IoT devices) with decision making in the selection of service consumer, and SLA Negotiation. The service providers register with the proposed global manager through parameters predefined by the Global Manager. Few of the parameters are identified from service request specifications of cloud service providers [3], [49] and a few are retrieved from the literature [45], [46], [47], [48]. The first task of the global manager is to group and position the service providers appropriately. Pre-grouping service providers upon registration can reduce the search space and reduce the time required for shortlisting service providers corresponding to every consumer request. The outdated capacity information with brokers lead to longer waiting time for querying the consumers and allocation as specified in [1]. Hence in the proposed approach, the registered service providers updates their resource capacity periodically with the Global manager thereby ensuring reduction in waiting time of service request. The second task is to select appropriate groups of service providers, for every consumer request. The third task is to shortlist the service providers from the selected groups. The fourth task is selecting one best-suited service provider among the shortlisted service providers. The fifth task is, in a situation where the global manager is unable to select one best provider, the Global Manager is smart enough to perform a service negotiation with the shortlisted providers and suggest modification in service parameters to make them best fit for the service consumer request.

The organization of the paper is as follows, Section II describes various work related to proposed approaches such as integrated environment, cloud broker, cloud service selection methodology, multi objective optimization, feature selection, supply demand optimization and league championship algorithm. Section III describes the architecture of the proposed global manager with its functionalities. Section IV provides implementation details with experimental results and inferences. Finally, the conclusion envisaged from the proposed work and few identified future work to be carried on the proposed work are discussed in Section V.

2. Literature Survey

Shreshth et. al., [31] proposed FogBus framework supporting platform-independent application execution and node-to-node interaction in IoT-Fog-Cloud environment with Blockchain for data integrity, protection and privacy but did not provide the methodology for selecting appropriate execution nodes in the environment. Wang et al. [15] expressed the necessity for combined cloud service and proposed an adaptive learning mechanism through which different service types will be combined into an integrated cloud service. A novel trust-enabled service composition model (TALMSC) for mobile cloud environments is proposed in [13] by Wenjuan et. al., which involve the entities - mobile cloud users, service providers, and the service intermediater generally termed as broker. The new trust mechanism is context-aware and combines direct trust with the recommendation of trust. Dan et. al., [1] proposed a generic cloud brokerage architecture with B cloud-tree grouping, which is like service requests received during the same time interval. But the queries received at varying irregular intervals with numerous service providers are also to be considered. Jrad et al. [4] developed a cloud broker system that analyzes the parameters of user requirements along with SLA to automatically select cloud services. The approach may be time consuming when the number of providers increases, since each service provider is to be scanned and compared with the user requirements at every instance. A cloud brokering architecture is proposed by Wenqiang et. al., [14] where Cloud Instance Type Selection Model (CITSM) is modeled as a multi-objective optimization problem, as a result the relationship between complete Pareto set and solution space is obtained. A Complete Pareto Set Generation Algorithm (CPSGA) generates a complete Pareto set of the cloud instance type selection schemes as specified. It is followed by the Optimal cloud instance type selection Scheme Screening Algorithm (OSSA) that selects one scheme from the complete Pareto set. Sabrina et. al. [17] proposed a fuzzy-based cloud plan assessment brokering service where the users specify requirements in natural language expression that are evaluated by fuzzy logic to avoid sharp boundaries between acceptable and unacceptable values. Zhenni et. al., [20] proposed an iterative double-sided auction scheme to compute resource trading for edge-cloud-assisted IoT in a blockchain network focusing on pricing but not on other functional parameters, where the broker solves the problem of determining computing resource trade and designs a specific price rule for buyers and sellers.

John et. al., presented focus+ context technique based on hyperbolic geometry for visualizing and manipulating large hierarchies. This technique assigns more display space to a portion of the hierarchy while still embedding it in the context of the entire hierarchy [6]. Wiem et. al., [40] proposed a broker based on evaluation of the fault severity impact. It decides on continuation of service consumption with the existing service provider or switches to a new provider but lacks focus on selecting the provider at first instance. Souranil et. al., [41] specified major problems the broker faces such as scalability due to various reasons, namely increase in number of users, resiliency, and single point of failure. It implies that the designing of broker architecture needs to be addressed. Teerawat et. al., [42] proposed a multi-objective particle swarm optimization (MOPSO) for the profit maximization of the broker with minimal response time and energy consumption. Yexi et. al., [43] proposed a Cloud Service Marketplace for cloud-based appliances with Services Knowledge Graph leveraging data mining technology to enable semantic understanding of customer requirements. Lie et. al., [32] proposed a context-aware cloud service selection model involving two assessment method named subjective assessment and objective assessment. The approach is based on the comparison and aggregation. The first subjective assessment is extracted from cloud user feedback and the later objective assessment is based on quantitative performance testing.

Based on the context similarity between the two assessment approaches, a filtering process is involved. The objective assessment depends on third party testing or performance measures. The reliability of such a third party is a critical consideration in this approach. Zia et. al., [30] proposed a multi-criteria cloud service selection methodology using a mathematical form of service selection, effective among providers with similar service specifications, but this is generally not applicable in a heterogeneous provider specification scenario. Xiaogang et. al., [24] proposed a dynamic cloud service selection strategy that used an adaptive learning approach with incentive and forgetting function. This approach selects the optimizing service dynamically and returns the integrated solution to users. This approach employs an intermediary service agent (ISA) model providing a parallel selection and an adaptive learning mechanism to support DCS selection. PSO Non-dominated Genetic Algorithm (PSONSGA) is proposed in [19] to solve the multi-objective optimization problem at fog level. Hybrid Artificial Bee Ant Colony Optimization (HABACO) load-balancing algorithm is proposed to improve the performance of the cloud in cloud-fog environment. The selection is based on service broker policies. Kalyanmoy et. al., [12] describes the NSGA II, where the combined population of parent and offspring is sorted and the best N members are chosen among the population. Then by selecting each non-domination level, one at a time through a computationally efficient niche-preservation operator, a new population is constructed. For every member, a crowding distance is calculated as the sum of objective-wise normalized distance between two neighboring solutions. Kalyanmoy et. al., [10] proposed a reference point-based approach to the NSGA-II framework for solving many-objective optimization problems to find near Pareto-optimal solutions corresponding to the supplied reference points. These reference points can either be predefined in a structured manner or supplied preferentially by the user. The crowding distance in NSGA II is replaced by the reference point in NSGA III. This NSGA III is enhanced with multiple reference points and employed in the proposed global manager. The authors in [18] propose an improved evolutionary multi-objective optimization algorithm combining fast non-dominated sorting of NSGA II and the genetic operators of grouping genetic algorithm. The strategy proposed by the authors in [24] to support DSC selection, provides an adaptive learning mechanism and parallel selecting means. The mechanism runs inside each ISA and involves incentive and forgetting functions that can realize the self-adaptive regulation for optimizing next service selection according to current service selection status. A fuzzy rough set-based feature selection algorithm was proposed by Hong et. al. [27] based on the complicated data structure of modern datasets, for large-scale hierarchical classification. The fuzzy lower approximation is computed with the sibling strategy to evaluate the features. This considers only the tree structures of class labels. Filter, wrapper and embedded methodologies are outperformed by the hybrid feature selection [23]. Feature selection strategy eliminates redundant data and extracts informative related data. The feature information of all attributes is replaced with effective information in the selected feature subset. Metaheuristic optimization algorithm achieves speed, robustness, and accuracy close to the optimal solution [26]. Hence a hybrid feature selection strategy is employed in the proposed approach. Feature selection is a preliminary step in the classification of data with higher dimensions. Feature selection aims to select a subset of features to decrease the time complexity, reduce the storage burden, and improve the generalization ability of classification [27]. Feature selection is gaining significance in research on multiple domains for extracting the most useful and valuable information, and to deal with high-dimensional mass data [28]. The optimization process is to obtain an optimal solution under certain constraints from the available solution space by way of minimizing or maximizing the objective function [33]. Literature [16] suggests that, among the three approaches such as Multi-Criteria

Decision-Making based approaches, Optimization-based approaches and Logic-based approaches, the optimization-based approaches are the most related to the cloud service selection approach.

Weiguo et. al., [33] proposed supply-demand-based optimization (SDO), a swarm-based optimizer. This metaheuristic optimization algorithm is motivated by the supply-demand mechanism in economics. As per the supply-demand optimization [33], in the market, when the current commodity price increases, commodity quantity will also increase in supply at the next time. When the commodity quantity increases, its price will decrease. The League Championship Algorithm LCA is considered an effective algorithm for resource allocation, resource management, resource distribution, search technique, and scheduling [34]. The League Championship Algorithm proposed by Kashan [35] is based on the sports league competition of sports teams as a swarm optimization problem with potential efficiency in solving many optimizations, non-deterministic polynomial-time (NP-complete) and real-world problems. The teams devise a new solution in the form a formation strategy using a metaphorical SWOT analysis based on their current solution. LCA is proposed as the stochastic population-based algorithm for continuous global optimization [36]. Shafi'I et. al. [37] proposes a Check-pointed League Championship Algorithm (CPLCA), a task scheduling with fault tolerance using the check-pointing strategy and the migration of tasks during unexpected independent task execution failure. The algorithms proposed are modified or enhanced algorithms retrieved from the literature such as for grouping the service providers is the modified version of radial tree algorithm by Nihar et. al. [8]. The proposed transversal is modified version of Book, G. [7] et. al. The proposed service provider selection is performed by the modified version of LWP algorithm proposed by F. Luo et. al. [9]. The proposed algorithm for short listing is a modified algorithm with combination of multiple feature-ranking methods proposed by Anwar et. al., [29] and context-aware cloud service selection framework proposed by Lie et. al. [32]. The proposed provider finalization algorithm is enhancement to reference point based multi objective evolutionary algorithms by Kalyanmoy et. al. [10].

3. Proposed Architecture

The proposed Global Manager is designed as an independent service to interoperate with maximum extent of possible service providers. The integrated environment with the proposed Global Manager is depicted in Fig. 1.

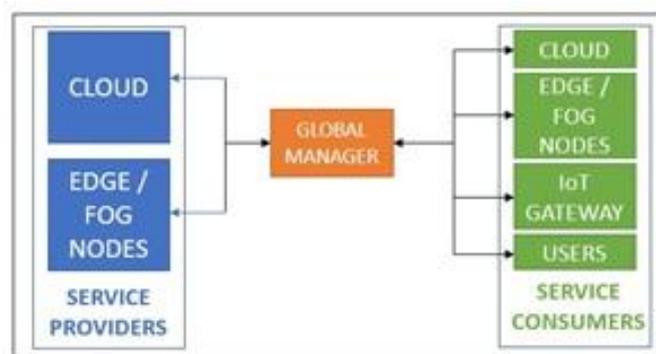


Fig. 1. Global Manager in Integrated environment.

The Global Manager comprises the following entities in the proposed scenario:

- **Service consumers** include cloud environment, Edge / Fog nodes, IoT gateway and users. The consumers submit their requirements to the proposed global manager. As the utilization and demand for cloud services, single cloud (as a cloud service provider) generally finds it difficult to serve its consumers. To improve the serviceability of cloud to its demanding customers, these clouds integrate with certain other cloud service providers for the need of IaaS / PaaS / SaaS or for cloud management services such as back or disaster recovery. This situation leads to the strategy of Hybrid cloud. Hence considering the situation, the cloud environment is also positioned in the service consumer category.
- **Service providers** comprise cloud environment (public, private, hybrid) and edge nodes. It is considered that edge nodes are managed by its own gateway.
- **Global manager** receives the request from the consumers, analyzes and processes the request to identify the best service provider corresponding to every request.

The operational considerations of the proposed global manager are as follows:

- The service providers should register with the global manager for offering their services through the integrated environment.
- The parameters for the registration of service providers are predefined by the global manager.
- The service providers, on successful registration are expected to update their resource availability status with the global manager periodically. The service provider generally assures the global manager with minimum and a maximum capacity of service at any instance.
- The service provider should be ready to undergo certain changes in their service offering based on service negotiation by the global manager to make them best fit for any incoming request, if necessary.
- The service consumer should register with the global manager to avail service from the integrated environment.

The architecture of the proposed global manager is depicted in [Fig. 2](#). The components of the proposed global manager are categorized into the following major components:

- Registration Module
- Service Provider Grouping Module
- Service Provider Group Selection Module
- Service Provider Short listing Module
- Service Provider Finalization Module
- Service Negotiation Module

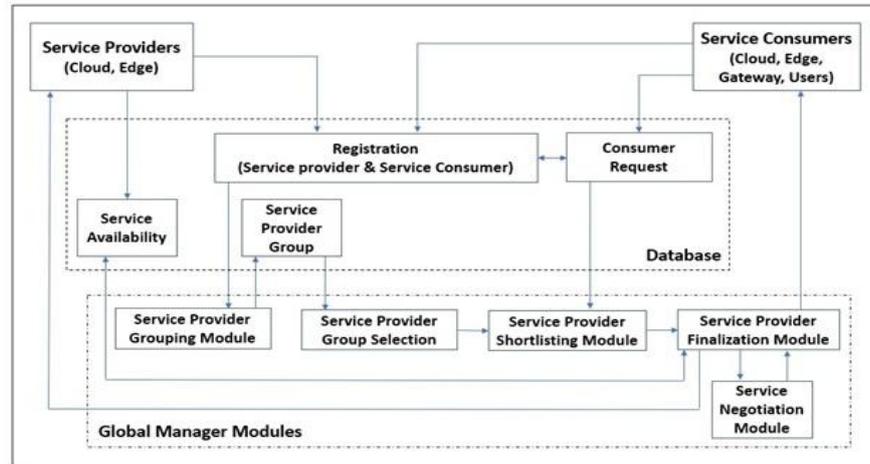


Fig. 2. Architecture of the Global Manager.

The workflow of the proposed global manager components is depicted in Fig. 3.

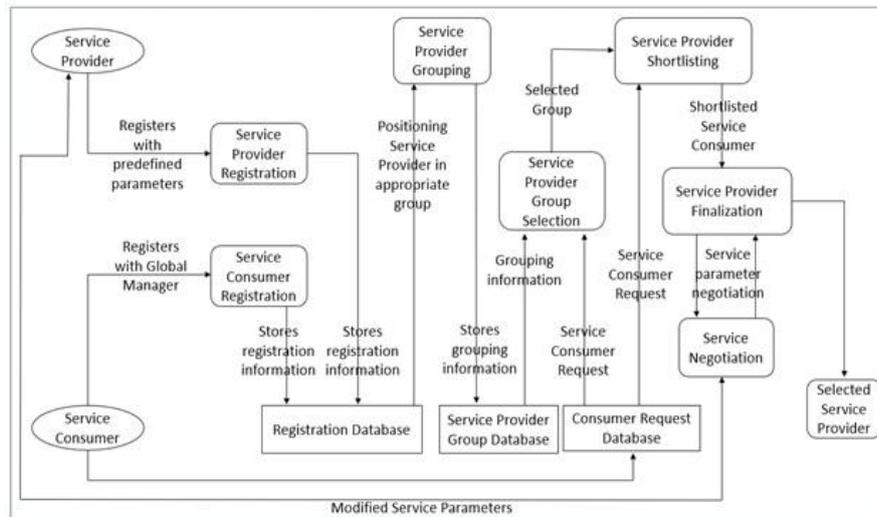


Fig. 3. Workflow with the Global Manager Components.

The workflow is as follows:

- All the Service Providers register with the Global Manager with the parameters predefined by the Global Manager. These parameters define the services offered by the providers. The registration information is stored in the Registration Database.
- Based on the parameters predefined by the Global Manager, various Service Provider groups are formed and upon registration by the Service Provider, the providers are positioned in the appropriate group. This function is performed by the Service Provider Grouping Module. This grouping information is stored in the Service Provider Database.
- Service Consumers register with the Service Consumer Registration module. This registration information is stored in the Registration Database.
- When a Service Consumer requests for a service with Global Manager, the request is stored in Consumer request database.

The request is processed in the following steps:

- Step 1: Lists of service provider group / groups are selected among the available service provider groups based on the Service Consumer request with few of the service consumer request parameters (Parameter set 1).
- Step 2: From the matched service provider group / groups, appropriate members of the group (service providers) are shortlisted corresponding to the service consumer request parameters (Parameter set 2). This step is performed by the Provider Short listing Module.
- Step 3: Among the shortlisted members (service providers), one best service provider is selected by the Service Provider Finalization module.
- Step 4: In a situation where one best service provider cannot be identified, the Service Negotiation Module performs a service negotiation function. The value of service parameters can be varied by the Service providers to make them the best bit for that particular consumer request. Upon variation in the value of service parameters, step 3 is performed to find the best matching service provider.

3.1 Registration Module

3.1.1 Service Provider Registration

The service providers register the specification of the parameters of their services being offered with service provider registration service. The set of parameters for registration are predefined by the Global Manager. In the proposed approach 30 parameters are considered for service registration. However, the number of parameters can be customized by scaling up or down. These parameters are organized into three sets based on the importance of the parameters to the consumers as explained in the following paragraph.

Parameter set – 1 is generally considered for primary categorization of the service providers (like public Cloud / private Cloud service provider and so on.) and forms the base for the formation of the grouping of service providers as explained in the next section. Parameter set – 2 forms the next base of requirements of a consumer. These are important parameters and would be required to fulfill the request of the consumer. Parameter set – 3 are flexible parameters and are used for the purpose of negotiation at a later stage provider selection or finalization.

- Parameter set 1

Five parameters are categorized in set -1. These parameters are of multiple choice type. The parameters listed are as follows:

- Service provider category - Cloud or Edge,
- Type of service provider - Public or private,
- Virtualization technology – Hypervisor or Container or Bare metal,
- Availability of reservation facility - Yes or No,
- Availability of platform service - Yes or No.

- Parameter set 2

Twenty out of remaining twenty five parameters are positioned in set – 2. The parameters are as follows: size and structure of each VM with its memory, vCPU and storage; minimum and maximum no. of VMs of each size; operating system available; storage type available (SSD/ Sata/ Sas); cost/VM/month based on size; backup type and category; availability of high availability feature with no. of VMs in each cluster; availability of elastic instance with max. no of elastic instance; service availability schedule; availability of security service like firewall, directory service; networking services available like DNS, IP scheme, load balancer;

APIs available; storage security mechanism offered like encryption, keypair support by self / third party; geographical location of data centre; Category of data centre (Tier 1 or 2 or 3); level of support offered for the service; availability of boot from volume; category of identity service. The parameters set 2 acts as the input for the provider shortlisting module, with details explained in the forthcoming section.

- Parameter set 3

Five numerical based parameters are positioned in set – 3. The parameters are as follows: cost of service based on capacity / month; minimum and maximum bandwidth; service availability (based on SLA) in percentage; recovery time objective and recovery point objective for service recovery; cost for high availability, elastic instance and backup instance. These act as input for the provider finalization module and service negotiation module.

3.1.2 Service Consumer Registration

The service consumers as depicted in [Fig. 1](#), (Cloud service providers, Edge nodes, gateway and users) who wish to avail the service of Global Manager registers with the Global Manager.

3.2 Database

The following databases are considered in the proposed Global manager environment: Registration (Service provider & Service consumer), Consumer request, Service provider group, Service availability, Service availability update.

3.3 Service Availability Module

The service providers upon registration with the Global Manager, specifies the minimum and maximum capacity that can be used for allocation by Global Manager. The upfront specification of minimum and maximum capacity helps the Global Manager in capacity planning at a faster pace by keeping track of the resource availability from all services. The maximum resource capacity is denoted by {max.cap} and minimum resource capacity is denoted by {min.cap}. On allocation of service requests to providers, the request gets provisioned as virtual machines. Due to the auto scaling feature of Cloud, the consumer workload can consume more resources based on the resource policies of the service provider.

Hence, service providers update their resource availability denoted by {serv.aval} which is of $\text{min.cap} < \text{serv.aval} < \text{max.cap}$. The service capacity is verified as

$$\text{ServProv.temp_serv.aval} = \text{max.cap} - \{\text{Globalmanager.alloc.cap}\}.$$

$$\text{ServProv.temp} = \text{select if ServProv.temp_serv.aval} > \text{consumer_request}. \quad (1)$$

The min.cap is the minimum service capacity that service providers commit to Global Manager and max.cap is the maximum service capacity that service providers commit to Global Manager. The max.cap value is dynamically variable by the service providers. Serv.aval is the current service availability capacity available with the service provider which is updated to the Global Manager more frequently. Serv.Prov.temp_serv.aval is the service availability capacity available with the shortlisted service provider. The shortlisted service provider is selected for further level of filtration if the shortlisted service provider service availability is greater than the consumer request capacity. The frequency of update is proportional to the change in the resource consumption. This value of serv.aval helps the Global Manager to make an instant decision on the short listing of the service provider based on resource capacity available with every service provider.

3.4 Service Provider Grouping Module

With the parameter set 1 categorized in service provider registration, all possible groups are formed by the global manager with the proposed service provider grouping algorithm. Upon registration of service providers, the service providers are positioned in appropriate groups by the Global manager by mapping the value of parameter set 1 specified by the service providers to appropriate groups. The Service provider grouping module performs two major functions such as Group formation based on service provider grouping algorithm and positioning service providers upon registration in appropriate groups.

3.4.1 Service Provider Grouping Algorithm

In the proposed approach, each parameter to be considered for formation of groups are positioned in consecutive concentric circles also called as rings, eliminating the parent-child bonding. The position of rings in the structure is interchangeable without any changes in the grouping structure and the number of rings in the structure can be scaled up or scaled down dynamically. The parameter set 1 considered during the service provider registration is employed for group formation. The positioning of parameters in consecutive rings is selected in random. A node is a position in a ring, possessing certain property predefined by the Global Manager. Changes in the properties of nodes will not impact the group formation. All the rings are considered as peers and positioned in sequential order. In the proposed approach, the organizations of rings corresponding to each parameter with its properties are as follows:

- Parameter (ring 1) - Service provider category - Cloud / Edge
- Parameter (ring 2) - Type of service provider - Public / Private
- Parameter (ring 3) - Virtualization technology - Hypervisor / Container / Bare metal
- Parameter (ring 4) - Availability of reservation facility - Reservation / No reservation
- Parameter (ring 5) - Availability of platform service - Platform / No Platform

The structure of the groups formed by the proposed approach is depicted in the following

Fig. 4.

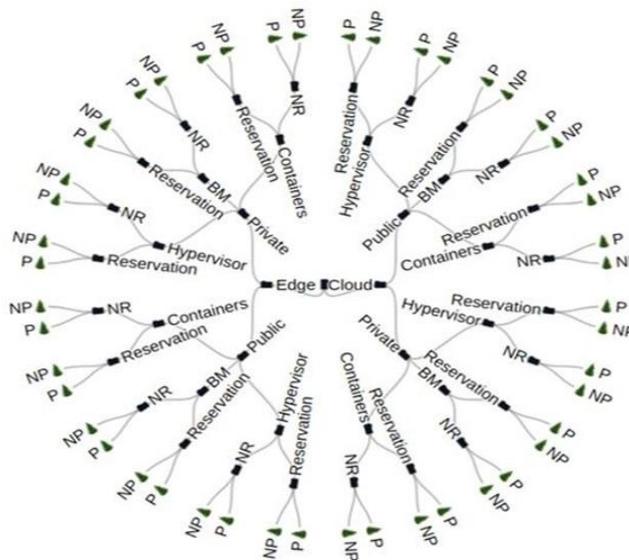


Fig. 4. Provider grouping structure

All the parameters are positioned along the arc of ring. Parallel lines emerge in hyperbolic structure with each successor parameter will see an arc that spans at an angle as its predecessor. There is no overlap on the arc of the successor parameters.

The grouping structure is represented by the following expression (2),

For initiating node,

*Angular width = (angular width of center node / no. of nodes at level two) * no of subsequent nodes, if non-leaf*

= angular width of center node / no. of nodes at level two, if leaf node.

For nodes of level other than level one,

Angular width = (angular width of initial node) / no. of parameters corresponding to the node

*Angular degree = Initial node's angle – initial node's angular width / 2 + node's angular width * node's index.* (2)

To position service providers in appropriate groups, the value of every parameter provided during the registration process is mapped onto corresponding rings through the proposed transformation function. When a member is to be assigned a new position in the succeeding ring, a transformation is computed which positions indicated point on the hyperbolic plane. The layout transverse through a hyperbolic radial tree to compute the successor ring position with reference to the predecessor position. The vertex is represented as p, midline endpoint represented as m, and angle of the arc, with Trans as transformation function, a transformation apparatus is calculated. This function returns a transformation point with input as point and transformation specification.

$$\{P = \text{Transver}(m, \langle p, 1 \rangle), m' = \text{Transver}(\text{Transver}(m, \langle p, 1 \rangle), \langle -p', 1 \rangle), a' = (\log(\text{Transver}(e, \langle -d, 1 \rangle)))\}. \quad (3)$$

In the proposed transversal, any given member who does not possess a positive value for any of the parameters pointing to a ring, then the members will be positioned in a false value for a Boolean value parameter and in none of the positions in the ring for other values. This transverse is rigid which is expressed as a complex function of z of the form specified in (4)

$$z' = \frac{P + m'}{1 + a'} \quad (4)$$

3.5 Service Provider Group Selection Module

Upon receipt of a service request from Service consumers, appropriate group / groups of consumers are selected by the Service provider group selection module. Based on the request specification corresponding to the category – 1 under registration, submitted by the service consumer when raising a service request, appropriate group / groups are selected. In the proposed algorithm, if no value is specified for a parameter in request specification, then groups with all values in the ring are selected. The proposed algorithm selects the group based on the modularity score. All the groups with modularity score $M \geq 1$ are selected. The Matching modularity is provided by M in (5) where the groups are represented as nodes.

$$M = \frac{2^1 \times \sum_{ij} A_{ij} \theta(i, j)}{\sum_{ij} \lambda(i, j)} \quad (5)$$

The outcome of this module is the list of selected group / groups corresponding to the service consumer request.

3.6 Service Provider Group Short Listing Module

On selecting the appropriate group of service providers for every service request, the service providers are shortlisted from the selected group / groups. The parameters set 2 classified in

the registration process are considered for the service provider short listing process. The objective and subjective parameters are normalized to trapezoidal fuzzy numbers in the context-aware cloud selection framework. The processing is through the following steps,

- Step 1 (Conversion of subjective and objective attributes values into ratings): The subjective, objective and quantitative values in parameter set 2 are converted into trapezoidal fuzzy numbers and these numbers are further converted as fuzzy ratings on comparison with every fuzzy value to the best fuzzy value of each parameter.
- Step 2 (Filtering out assessments): For each assessment, the Euclidean distance between the fuzzy ratings of the associated attributes is computed. If the distance exceeds a fixed threshold, the assessments offering such values of the subjective attributes are removed from the fuzzy rating matrix.
- Step 3 (Computing weight for each attribute): According to the requirement specification, a weight in the form of linguistic variables is assigned to each attribute.
- Step 4 (Aggregating all attributes): After filtering in Step 2, by the revised fuzzy rating matrix and the importance weights, final score for every service is computed. All the shortlisted services are ranked for selection.

The multiple feature-ranking method identifies relevant features and eliminates redundant features. The features which are ranked lower than a certain threshold is filtered-out. Using linear correlation coefficient, association between data point and corresponding value are measured.

$$w_i = \frac{d(w_{s_i})}{\sum_{i=s+1}^{s+o} d_i}, \quad R_v = (1 - w_i) \times \quad (6)$$

The outcome is the list of shortlisted service providers from various matching groups.

3.7 Service Provider Finalization Module

The service provider shortlisted in the service provider shortlisting module is processed by the service provider finalization module to select the best suitable service provider. The finalization is based on the parameter set – 3 categorized in the registration. The parameters to be considered for finalization are positioned in a normalized hyper plane. In the proposed algorithm, every normalized hyper plane will have its ideal point formulated by the global manager from individual consumer requests. Corresponding members are positioned on the plane over the reference points. Ideal point represents the value of every parameter in user request. The ideal point in all planes will form an ideal plane. The members positioned in the ideal plane will be considered for selection. The members with closest reference point distance termed as ideal value from all planes are considered for finalization. The normalization objective function is defined in (7).

$$H = M \frac{+p-1}{p} f_i^n(x) \text{ where } f_i^n(x) = \frac{f^i(x)}{a_i} \quad (7)$$

In the proposed approach, 5 objectives are considered hence a five dimensional plane is depicted in the following **Fig. 5**.

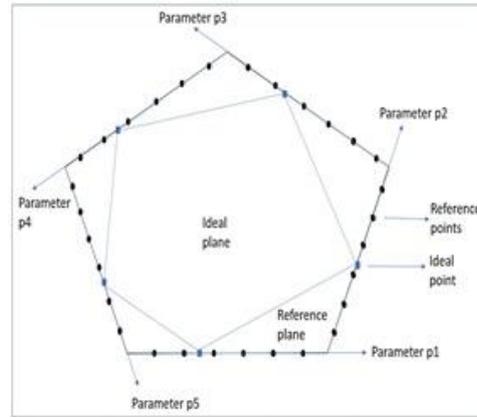


Fig. 5. Ideal points in each plane

The ideal value is the difference between the ideal point and the corresponding member's value for every parameter. The ideal value can be either positive or negative to any of the objectives. On obtaining the ideal value for all considered population members, a ranking mechanism is applied. The shortlisted member with maximum positive ideal value is positioned with the highest ranking and the one with maximum negative ideal value is considered with lowest ranking.

The following scenarios are considered in the finalization of the service provider:

- If the count of service providers with Ideal value of all planes are positive is one, the member (service provider) is finalized.
- If the count of shortlisted members (service providers) with Ideal value of all planes is positive is more than one, a ranking function is applied to all shortlisted members. The ranking function is based on the minimum and maximum ideal value corresponding to each objective. The top ranked service provider is finalized.
- If no service provider with all positive values is found, it means that the provider finalization module is not able to finalize a single service provider. In such cases, all the selected providers are generally subjected to undergo a service negotiation process governed by the service negotiation module. The service negotiation process returns the negotiated value of the service provider whereby step 1 and step 2 are followed.
- Even after the service negotiation process, no appropriate service provider is finalized; the service request is held in the queue.

The outcome of the provider finalization process, an appropriate service provider is finally selected for a corresponding service consumer request.

3.8 Service Negotiation Module

In a situation where the service provider selection module is unable to select the best service provider, a service negotiation process is involved. In the service negotiation process, the selected service providers with the nearest possible degree of match are subjected to undergo modification in level -3 parameters of service offering. The service providers who modify their service functionality to have the best degree of match based on the suggestion of the Global Manager are considered for final selection. A modified Supply-Demand-Based Optimization algorithm is proposed for the service negotiation process. The supply function q and demand function p are expressed as defined in (8)

$$q_{t+1} - q_0 = a(p_t - p_0) \text{ and } p_{t+1} - p_0 = b(q_{t+1} - q_0) \quad (8)$$

Upon selection of a single service provider, the service request under consideration is forwarded to the finalized service provider. If the Global Manager is unable to find an appropriate service provider even after the service negotiation process, that service consumer request is kept on hold in the request receiving module.

The overall Global Manager functioning is explained as

Input :

Service_consumer_request {with Parameter 1~n} where n is max. 30 in the proposed paper.

Service_providers_registration {Parameter 1~n} where n is max. 30 in the proposed paper.

Service_provider_Capacity_service_availability represented by serv.aval.

Initialize Global_Manager

Step 1 : Group formation by Service_Provider_Grouping_Module

Identify number of parameters for group formation m where m is subset of n.

Radial_tree structure: Position every parameter in subsequent ring of grouping structure

Identify number of nodes in every ring and position nodes in rings with angular_width and angular_degree.

Step 2 : Registration of Service_Providers and Service_Consumers

Input: Service consumer request with input parameter 1 ~ n.

Function: Store Service_Providers_Registration in Database

Step 3 : Positioning of Service_Providers in group by Service_Provider_Grouping_Module

While Service_provider=Not Null

For i = 1 to n,

Transverse Horizontally and vertically by Z', where horizontal is transversal to nodes in same ring level and vertical is transversal to rings in subsequent level.

Step 4 : Service_Provider_Group_Selection

Input: Service provider groups and Consumer request parameter values

For every consumer request, Consumer_request_parameter = j where $j \leq n$ and $n=30$ in proposed paper,

Select m (value <1~5>) from j where m is parameter governing the group formation

For m (value) -> Select matching Service_Provider_Groups governed by matching modularity M.

Output: Selected_Service_Provider_Groups.

Step 5 : Service_Provider_Shortlisting

Input : Members (Selected_Service_Provider_Groups) & Consumer_request_value (m')

From Selected_Service_Provider_Groups, Shortlist Service_Providers governed by Service Provider Shortlisting Module.

Output : Shortlisted_Service_Providers

Step 6 : Service_Provider_Finalization

Input : Shortlisted_Service_providers (k) & Consumer_request_value (m'')

For all members (k),

Check Matching service provider governed by

Service_Provider_Finalization_Module with H.

{

If all Value (m'') satisfies any member (k), select kn && if Serv.aval (kn) > value (Consumer_request (m))

then finalize kn.

else invoke Service_negotiation_module if not round 2.

Output : Selected service provider if all value of (m'') matches, else Step 7.

Step 7 : Service_negotiation

Input : Selected_service_providers k , from Step 6.

Variation in value of service offering Modified Value(m''')

Output : Modified Value (m''') for every member k .

Go to Step 6 as round 2.

Output : Finalized Service_provider

4. Implementation, Experimentation, Results and Observation

The cloud environment is implemented with OpenStack [57] on three nodes, VMware on three nodes and Cloudsim on a single node. Two nodes are configured as Edge computing node with StarlingX [25], an OpenStack based Edge computing framework. The Global Manager is implemented with Python and MySQL. The services of Global Manager are exposed as web services for ease of integration and interoperability in a heterogeneous environment. The formation of grouping structure governed by the service provider grouping module is dynamic in nature. The group is initially formed with three grouping parameters and scaled up to four and finally with five grouping parameters as specified in earlier sections. Moreover, it is observed that the positions of the rings within the groups are interchangeable without deviation in positioning the members to be placed in corresponding rings. Hence the position of members in corresponding groups will be face any deviation while changing the grouping structure by interchanging the position of rings and scaling up the number of rings in the groups. Also, it is observed that the time taken for formation of groups is negligible even with increase in the number of parameters used for group formation, in other words increasing the number of rings in the grouping structure. The below mentioned **Table 1** specifies the time taken for formation of groups and positioning of members in corresponding groups.

Table 1. Grouping information

S. No.	No. of rings in group	No. of parameters in each ring					No. of service providers	Time in seconds	
		Ring 1	Ring 2	Ring 3	Ring 4	Ring 5		For group formation	For positioning all members within groups in sequence
1	3	2	3	2	0	0	25	0.06	0.11
2							50	0.06	0.14
3							75	0.07	0.13
4							100	0.07	0.15
5							125	0.07	0.15
6							150	0.08	0.16
7	5	2	2	3	2	2	25	0.07	0.14
8							50	0.08	0.14
9							75	0.07	0.16
10							100	0.09	0.17
11							125	0.10	0.19
12							150	0.10	0.20

Execution results specified in **Table 1** is represented in graphical form in **Fig. 6** and **Fig. 7**.

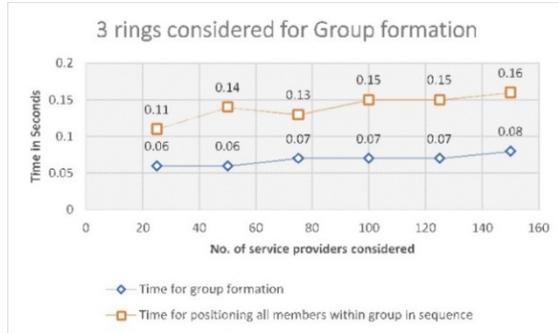


Fig. 6. Three parameters for Group formation

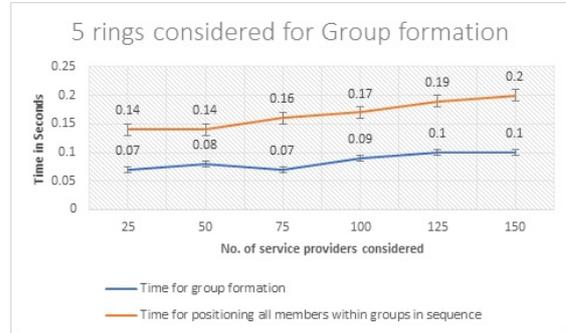


Fig. 7. Five parameters for Group formation

The required dataset is retrieved from the literature [1], [2], [5], [17], [23], [51], [52], [53], [54], [55], [56], [58] and from the cloud portals of corresponding service providers [3], [49], [50], [59]. Few of the data is generated using random generator function. The partial code for implementation is retrieved from the existing work [11], [22] and customized to perform the proposed functions.

The proposed algorithm is implemented with the specified data sets. The execution is done with varying request parameters ranging from single consumer request with 20 request parameters to a maximum of five consumer request simultaneously comprising of 30 parameters / consumer request. This request is provided to the Global Manager for execution and the time taken for execution is provided in Table 2.

Table 2. Time for execution of request through proposed Global Manager

S. No.	No. of parameters considered as request	No. of service providers considered	Time for execution of proposed approach in Seconds	Time for execution of normal query in Seconds
1	Single Consumer request with 20 parameters	25	0.25	0.03
2		50	0.61	0.09
3		75	1.12	0.22
4		100	1.16	0.50
5		125	1.21	0.77
6		150	1.57	0.84
7	Single Consumer request with 30 parameters	25	0.23	0.01
8		50	0.37	0.02
9		75	0.61	0.04
10		100	0.81	0.15
11		125	1.40	0.42
12		150	1.64	0.91
13	Five Consumer request with 20 parameters / request totaling to 100 parameters under consideration	25	0.37	0.05
14		50	0.80	0.11
15		75	1.17	0.30
16		100	1.59	0.70
17		125	1.70	1.11
18		150	2.25	1.13
19	Five Consumer request with 30 parameters / request totaling to 150 parameters under consideration	25	0.28	0.02
20		50	0.56	0.03
21		75	0.85	0.06
22		100	1.01	0.19
23		125	1.90	0.60
24		150	2.06	1.19

Execution results specified in **Table 2** is represented in graphical form in **Fig. 8**.

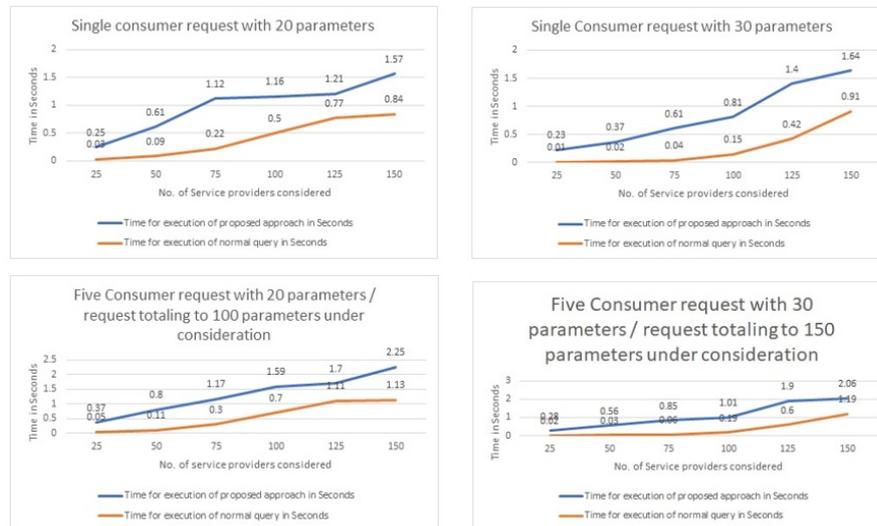


Fig. 8. Consumer request Vs. Execution

It is observed that although the time taken for execution of the consumer request through the proposed approach is considerably more than the conventional querying mechanism, the time difference is negligible. The proposed approach includes the service negotiation framework, which increases the chances of the consumer request being served and reduces the service provider resources time being idle thereby increasing the utilization of service provider's services. Hence the proposed approach brings in mutual benefit to both service consumer and service provider, on ignoring the time for identifying the appropriate service provider. On analysis of the execution, it may be noted that as the number of service providers, number of parameters and number of requests increases, the difference in time taken for proposed approach and conventional approach reduces and the difference can be ignored.

4. Conclusion and Future Work

The proliferation of Cloud services across various domains increases the demand for cloud services. Due to the demand, various cloud services are evolving with various transformations in service like Edge / Fog computing leading to an integrated environment. The domain requirements are also heterogeneous in nature which is difficult for any service consumer to find an appropriate service provider. A service broker is highly essential to find an appropriate service provider in an integrated environment corresponding to the heterogeneous requirements of service consumers. Hence the Global Manager is proposed in the paper which serves the integrated environment comprising of cloud, edge / fog nodes and IoT environment. The Global manager is implemented as web services in a way that it is interoperable with any standard service providers. The experimentation reveals that the proposed Global manager is effective in execution, thereby effectively serve the consumers. The pre-grouping of service providers helps the Global manager in finding the appropriate service providers with a minimal query interval. The approach of ideal points along every plane in the service provider finalization module makes the task of identification of appropriate parameter for service

negotiation much easier, thereby ensuring the service provider as best fit to the service consumer. This negotiation also increases the utilization of appropriate service utilization.

The proposed Global Manager can be extended with various functionalities including automated evaluation of service level agreement of service providers, mechanism to monitor and track the SLA adherence by service providers, service arbitration by Global Manager, consideration of security and access control mechanism between service providers and consumers, orchestration and resource management including monitoring in an integrated environment, consideration of interoperability and portability among service providers federated by the Global Manager.

Acknowledgement

The experimentation is performed in the Cloud computing environment in Centre for Development of Advanced Computing, Chennai.

References

- [1] Dan Lin, Anna Cinzia Squicciarini, Venkata Nagarjuna Dondapati, and Smitha Sundareswaran, "A Cloud Brokerage Architecture for Efficient Cloud Service Selection," *IEEE Transactions on Services computing*, Vol. 12, No. 1, pp. 144-157, January / February 2019. [Article \(CrossRef Link\)](#)
- [2] <https://cloud.google.com/>
- [3] <https://aws.amazon.com/>
- [4] F. Jrad, J. Tao, A. Streit, R. Knapper, and C. Flath, "A utility-based approach for customised cloud service selection," *International Journal of Computer Science and Engineering*, Vol. 10, no. 1/2, pp. 32-44, Jan. 2015. [Article \(CrossRef Link\)](#)
- [5] A. Hammoud, H. Otrok, A. Mourad, O. Abdel Wahab and J. Bentahar, "On the Detection of Passive Malicious Providers in Cloud Federations," *IEEE Communications Letters*, Vol. 23, no. 1, pp. 64-67, Jan. 2019. [Article \(CrossRef Link\)](#)
- [6] Jonh Lamping, Ramana Rao, "The Hyperbolic Browser: A Focus+Context Technique for Visualizing Large Hierarchies," *Journal of Visual Languages & Computing*, Vol. 7, no. 1, pp. 33-55, 1996. [Article \(CrossRef Link\)](#)
- [7] Book, G. and Keshary, N., "Radial tree graph drawing algorithm for representing large hierarchies," *Technical report, University of Connecticut*, 2001.
- [8] Nihar Sheth and Qin Cai, "Visualizing MeSH Dataset using Radial Tree Layout," *Indiana University*, Bloomington, April 29, 2003, Corpus ID. 15920491.
- [9] F. Luo, J. Z. Wang, and E. Promislow, "Exploring local community structures in large networks," *Web Intelligence and Agent Systems: An International Journal*, vol. 6, no. 4, pp. 387-400, 2008. [Article \(CrossRef Link\)](#)
- [10] Kalyanmoy Deb and Himanshu Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577-601, Aug. 2014. [Article \(CrossRef Link\)](#)
- [11] <https://www.mathworks.com/matlabcentral/leexchange/71764-supply-demandbased-optimization>
- [12] Kalyanmoy Deb, Amrit Pratap, Samir Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, April 2002. [Article \(CrossRef Link\)](#)
- [13] Wenjuan Li, Jian Cao, Keyong Hu, Jie Xu and Rajkumar Buyya, "A Trust-Based Agent Learning Model for Service Composition in Mobile Cloud Computing Environments," *IEEE Access*, vol. 7, pp. 34207-34226, 2019. [Article \(CrossRef Link\)](#)

- [14] Wenqiang Liu, Pengwei Wang, Ying Meng, Qin Zhao, Caihui Zhao and Zhaohui Zhang, "A Novel Model for Optimizing Selection of Cloud Instance Types," *IEEE Access*, vol. 7, pp. 120508-120521, 2019. [Article \(CrossRef Link\)](#)
- [15] Xiaogang Wang; Jian Cao; Jie Wang, "A dynamic cloud service selection strategy using adaptive learning agents," *International Journal of High Performance Computing and Networking*, vol. 9, no. 1/2, pp. 70–81, 2016. [Article \(CrossRef Link\)](#)
- [16] L. Sun, H. Dong, F. Hussain, O. Hussain, and E. Chang, "Cloud service selection: State-of-the-art and future research directions," *Journal of Network and Computer Applications*, vol. 45, pp. 134–150, 2014. [Article \(CrossRef Link\)](#)
- [17] S. De Capitani di Vimercati, S. Foresti, G. Livraga, V. Piuri and P. Samarati, "A Fuzzy-Based Brokering Service for Cloud Plan Selection," *IEEE Systems Journal*, vol. 13, no. 4, pp. 4101-4109, Dec. 2019. [Article \(CrossRef Link\)](#)
- [18] C. Liu, C. Shen, S. Li and S. Wang, "A new evolutionary multi-objective algorithm to virtual machine placement in virtualized data center," in *Proc. of 2014 IEEE 5th International Conference on Software Engineering and Service Science*, Beijing, China, pp. 272-275, 2014. [Article \(CrossRef Link\)](#)
- [19] Z. A. Khan et al., "Energy Management in Smart Sectors Using Fog Based Environment and Meta-Heuristic Algorithms," *IEEE Access*, vol. 7, pp. 157254-157267, 2019. [Article \(CrossRef Link\)](#)
- [20] Z. Li, Z. Yang and S. Xie, "Computing Resource Trading for Edge-Cloud-Assisted Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3661-3669, June 2019. [Article \(CrossRef Link\)](#)
- [21] Shixiong Xia, Ranran Zhou, Yong Zhou, and Mu Zhu, "An Improved Local Community Detection Algorithm Using Selection Probability," *Hindawi Mathematical Problems in Engineering*, Vol. 2014, 2014, Article ID 406485. [Article \(CrossRef Link\)](#)
- [22] <https://github.com/glouwa/d3-hypertree>
- [23] Lin Sun, Tengyu Yin, Weiping Ding, Jiucheng Xu, "Hybrid Multilabel Feature Selection Using BPSO and Neighborhood Rough Sets for Multilabel Neighborhood Decision Systems," *IEEE Access*, vol. 7, pp. 175793-175815, 2019. [Article \(CrossRef Link\)](#)
- [24] Xiaogang Wang, Jian Cao, Jie Wang, "A dynamic cloud service selection strategy using adaptive learning agents," *International Journal of High Performance Computing and Networking*, Vol. 9, Nos. 1-2, pp. 70-18, 2016. [Article \(CrossRef Link\)](#)
- [25] <https://www.starlingx.io/>
- [26] Cao Han, Guo Zhou, Yongquan Zhou, "Binary Symbiotic Organism Search Algorithm for Feature Selection and Analysis," *IEEE Access*, vol. 7, pp. 166833-166859, 2019. [Article \(CrossRef Link\)](#)
- [27] Hong Zhao, Ping Wang, Qinghua Hu, Pengfei Zhu, "Fuzzy Rough Set Based Feature Selection for Large-Scale Hierarchical Classification," *IEEE Transactions on Fuzzy Systems*, Vol. 27, No. 10, pp. 1891-1903, October 2019. [Article \(CrossRef Link\)](#)
- [28] F. Nie, S. Yang, R. Zhang and X. Li, "A General Framework for Auto-Weighted Feature Selection via Global Redundancy Minimization," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2428-2438, May 2019. [Article \(CrossRef Link\)](#)
- [29] Anwar Ul Haq, Defu Zhang, He Peng, Sami Ur Rahman, "Combining Multiple Feature-Ranking Techniques and Clustering of Variables for Feature Selection," *IEEE Access*, vol. 7, pp. 151482-151492, 2019. [Article \(CrossRef Link\)](#)
- [30] Z. U. Rehman, F. K. Hussain and O. K. Hussain, "Towards Multi-criteria Cloud Service Selection," in *Proc. of 2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, Seoul, Korea (South), pp. 44-48, 2011. [Article \(CrossRef Link\)](#)
- [31] Shreshth Tuli, Redowan Mahmud, Shikhar Tuli, Rajkumar Buyya, "FogBus: A Blockchain-based Lightweight Framework for Edge and Fog Computing," *The Journal of Systems and Software*, 154, 22–36, 2019.

- [32] L. Qu, Y. Wang, M. A. Orgun, L. Liu and A. Bouguettaya, "Context-Aware Cloud Service Selection Based on Comparison and Aggregation of User Subjective Assessment and Objective Performance Assessment," in *Proc. of 2014 IEEE International Conference on Web Services*, Anchorage, AK, USA, pp. 81-88, 2014. [Article \(CrossRef Link\)](#)
- [33] Weiguo Zhao, Liying Wang, Zhenxing Zhang, "Supply-Demand-Based Optimization: A Novel Economics-Inspired Algorithm for Global Optimization," *IEEE Access*, vol. 7, pp. 73182-73206, 2019. [Article \(CrossRef Link\)](#)
- [34] Shafi'i Muhammad Abdulhamid, Muhammad Shafie Abd Latiff, Syed Hamid Hussain Madni, Osho Oluwafemi, "A Survey of League Championship algorithm: Prospects and Challenges," *Indian Journal of Science and Technology*, Vol 8(S3), pp. 1-10, February 2015. [Article \(CrossRef Link\)](#)
- [35] Ali. H. Kashan, "League Championship Algorithm: A New Algorithm for Numerical Function Optimization," in *Proc. of 2009 International Conference of Soft Computing and Pattern Recognition*, Malacca, Malaysia, pp. 43-48, 2009. [Article \(CrossRef Link\)](#)
- [36] Ali Husseinzadeh Kashan, "League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships," *Elsevier Applied Soft Computing*, Vol. 16, pp. 171-200, 2014. [Article \(CrossRef Link\)](#)
- [37] Shafi'i Muhammad Abdulhamid, Muhammad Shafie Abd Latiff, "A checkpointed league championship algorithm-based cloud scheduling scheme with secure fault tolerance responsiveness," *Elsevier Applied Soft Computing*, Vol. 61, pp. 670-680, 2017. [Article \(CrossRef Link\)](#).
- [38] P. Pawluk, B. Simmons, M. Smit, M. Litoiu and S. Mankovski, "Introducing STRATOS: A Cloud Broker Service," in *Proc. of 2012 IEEE Fifth International Conference on Cloud Computing*, Honolulu, HI, USA, pp. 891-898, 2012. [Article \(CrossRef Link\)](#)
- [39] Abdessalam Elhabbash, Faiza Samreen, James Hadley, and Yehia Elkhatib, "Cloud Brokerage: A Systematic Survey," *ACM Computing Surveys*, vol. 51, no. 6, pp. 1-28, Nov. 2019. [Article \(CrossRef Link\)](#)
- [40] Wiem Abderrahim, Zied Choukair, "Brokerage-based dependability integration in cloud computing services," *Journal of Super computing*, 74, 3359–3387, 2018. [Article \(CrossRef Link\)](#)
- [41] Souranil Sen and Aruna Balasubramanian, "A highly resilient and scalable broker architecture for IoT applications," in *Proc. of 2018 10th International Conference on Communication Systems & Networks (COMSNETS)*, Bengaluru, India, pp. 336-341, 2018. [Article \(CrossRef Link\)](#)
- [42] Teerawat Kumrai, Kaoru Ota, Mianxiong Dong, Jay Kishigami and Dan K. Sung, "Multiobjective Optimization in Cloud Brokering Systems for Connected Internet of Things," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 404-413, April 2017. [Article \(CrossRef Link\)](#)
- [43] Yexi Jiang, Chang-Shing Perng, Anca Sailer, Ignacio Silva-Lepe, Yang Zhou, Tao Li, "CSM: A Cloud Service Marketplace for Complex Service Acquisition," *ACM Transactions on Intelligent Systems and Technology*, Vol. 8, No. 1, pp. 1-25, Jan. 2017, Article no. 8. [Article \(CrossRef Link\)](#)
- [44] Yehia Elkhatib, "Mapping Cross-Cloud Systems: Challenges and Opportunities," in *Proc. of the 8th USENIX Conference on Hot Topics in Cloud Computing*, USENIX Association, pp. 77–83, June 2016.
- [45] Talal Halabi, Martine Bellaiche, "Towards Security-based Formation of Cloud Federations: A Game Theoretical Approach," *IEEE Transactions on Cloud Computing*, vol. 8, no. 3, pp. 928-942, 2020. [Article \(CrossRef Link\)](#)
- [46] Nikolay Grozev, Rajkumar Buyya, "Inter-Cloud architectures and application brokering: taxonomy and survey," *Software Practise and Experience*, 44, 369–390, 2014. [Article \(CrossRef Link\)](#)
- [47] Joonseok Park, Ungsoo Kim, Donggyu Yun, Keunhyuk Yeom, "C-RCE: an Approach for Constructing and Managing a Cloud Service Broker," *Springer Journal of Grid Computing*, 17, 137–168, 2019. [Article \(CrossRef Link\)](#)

- [48] Jingwei Li, Anna Cinzia Squicciarini, Dan Lin, Smitha Sundareswaran, and Chunfu Jia, “MMB^{cloud}-Tree: Authenticated Index for Verifiable Cloud Service Selection,” *IEEE Transactions on Dependable and Secure Computing*, Vol. 14, No. 2, pp. 185-198, March/April 2017. [Article \(CrossRef Link\)](#)
- [49] <https://azure.microsoft.com/>
- [50] <https://cloudharmony.com/>
- [51] Marco Guazzone, Cosimo Anglano, Matteo Seren, “A Game-Theoretic Approach to Coalition Formation in Green Cloud Federations,” in *Proc. of 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 618-625, 2014. [Article \(CrossRef Link\)](#)
- [52] Lena Mashayekhy, Mahyar Movahed Nejad, Daniel Grosu, “Cloud Federations in the Sky: Formation Game and Mechanism,” *IEEE Transactions on Cloud Computing*, Vol. 3, No. 1, pp. 14-27, January-March 2015. [Article \(CrossRef Link\)](#)
- [53] Alba Amato, Salvatore Venticinque, “Multiobjective Optimization for Brokering of Multi Cloud Service Composition,” *ACM Transactions on Internet Technology*, Vol. 16, No. 2, pp. 1-20, April 2016, Article no. 13. [Article \(CrossRef Link\)](#)
- [54] Consensus Assessment Initiative Questionnaire (CAIQ) v3.1 by Cloud Security Alliance [Online]. Available: <https://cloudsecurityalliance.org/artifacts/consensus-assessments-initiative-questionnaire-v3-1/>
- [55] Rajganesh Nagarajan, Ramkumar Thirunavukarasu, “A fuzzy-based decision-making broker for effective identification and selection of cloud infrastructure services,” *Springer Soft Computing*, 23, 9669–9683, 2019. [Article \(CrossRef Link\)](#)
- [56] Zahoor Ali Khan, Ayesha Anjum Butt, Turki Ali Alghamdi, Aisha Fathima, Mariam, Ariam Akbar, Muhammad Ramzan, Nadeem Javid, "Energy Management in Smart Sectors Using Fog Based Environment and Meta-Heuristic Algorithms," *IEEE Access*, vol. 7, pp. 157254-157267, 2019. [Article \(CrossRef Link\)](#)
- [57] <https://www.openstack.org/>
- [58] <http://beta-www.cloudcommons.com/web/cc/about-smi>
- [59] <https://deltacloud.apache.org/>



Kailash Selvaraj is working in Centre for Development of Advanced Computing (C-DAC), Chennai. His expertise includes Cloud Computing, Grid Computing and Cyber Security.



Dr. Saswati Mukherjee is working as a Professor in Department of Information Science and Technology, College of Engineering, Guindy, Anna University, Chennai. Her expertise includes Cloud Computing, Grid Computing, Text Mining, Information Retrieval, Operating System and Distributed Systems. She has published more than hundred research articles in International journals and International / National conferences, book chapters and successfully executed various research projects.