Pertanika J. Sci. & Technol. 28 (1): 117 - 140 (2020)

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

FLA-SLA Aware Cloud Collation Formation Using Fuzzy Preference Relationship Multi-Decision Approach for Federated Cloud

Pradeep Kumar Vadla[1]*, Bhanu Prakash Kolla[1] and Thinagaran Perumal[2]

[1]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, 520002 Vijayawada, Andhra Pradesh, India
[2]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

ABSTRACT

Cloud Computing provides a solution to enterprise applications in resolving their services at all level of Software, Platform, and Infrastructure. The current demand of resources for large enterprises and their specific requirement to solve critical issues of services to their clients like avoiding resources contention, vendor lock-in problems and achieving high QoS (Quality of Service) made them move towards the federated cloud. The reliability of the cloud has become a challenge for cloud providers to provide resources at an instance request satisfying all SLA (Service Level Agreement) requirements for different consumer applications. To have better collation among cloud providers, FLA (Federated Level Agreement) are given much importance to get consensus in terms of various KPI's (Key Performance Indicator's) of the individual cloud providers. This paper proposes an FLA-SLA Aware Cloud Collation Formation algorithm (FS-ACCF) considering both FLA and SLA as major features affecting the collation formation to satisfy consumer request instantly. In FS-ACCF algorithm, fuzzy preference relationship multi-decision approach was used to validate the preferences among cloud providers for forming collation and gaining maximum profit. Finally, the results of FS-ACCF were compared with S-ACCF (SLA Aware Collation Formation) algorithm for 6 to 10 consecutive requests of cloud consumers with varied VM configurations for different SLA parameters like response time, process time and availability.

Keywords: Cloud federation, collation formation, federated level agreement, fuzzy preferences relationships, key performance indicators, service level agreement

ARTICLE INFO

Article history:
Received: 11 May 2019
Accepted:13 September 2019
Published: 13 January 2020

E-mail addresses:
pradeepvadla@gmail.com (Pradeep Kumar Vadla)
drkbp@kluniversity.in (Bhanu Prakash Kolla)
thinaperug1@gmail.com (Thinagaran Perumal)
* Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

INTRODUCTION

Cloud computing is a computing paradigm which provides the solution to all enterprise applications requirements. The virtual machine (VM) plays a key role in providing computational resources at different instances of the request made by cloud consumers (Mishra et al., 2012). Most of the cloud consumers can be any enterprises of ranging from small scale to large scale using VM instances dynamically. Due to cloud computing on-demand and high scalability nature it needs for enterprises to face some unique challenges in terms of meeting the consumer's requirements for high data-intensive applications which require low congestion of resources and less rejection rate.

The best possible way to attain the above challenge is either avoid over-provisioning of resources for particular VM, but it lacks the efficiency of underutilized physical resources during low demand of consumer's request. Another approach as used by Amazon EC2 is to give authenticated user's service level guarantee for best services. Even in this differentiated model, the privilege users are facing request's rejection, unpredicted delays and resources shortage (Chen et al., 2011). One solution is, Federated cloud, a type of flavored cloud computing model which forms alliances among different service providers and provides resources to customers satisfying their different SLA needs (Rochwerger et al., 2011). In this model cloud providers will collaborate to form collation and try to meet all possible requirements of cloud consumers and satisfy their large demand of resources requirements by maximizing their price by providing good user experience (Rochwerger et al., 2009).

In this paper, we propose an FLA-SLA based cloud collation formation approach. In our scheme, each cloud consumer request consists of the number and type of resources with specified SLA requirement. The KPI's (Key Performance Indicator's) of different cloud providers are considered in this paper to form a Federated Level Agreement (FLA) to have a fine measurement of SLA to form collation at a particular instance. The set of KPI's which have been agreed among cloud providers are used for computing the SLA parameters for collated providers during collaboration and can be considered as FLA's and used as a reference to check for meeting the SLA of a user request. These SLA's are calculated at different instances and checked against user request of resources for collation formation by different cloud providers and provide choice for cloud consumers to choose collated providers who satisfy their request within a specified SLA limit. The main objective considered in our approach was to maximize the total profit gained by the collated cloud providers and share the profit based on their contribution of resources to the collation without FLA and SLA violation.

The state of art of cloud collation formation solutions in federated clouds from Mashayeckhy et al. (2014), Guazzone et al. (2014) and Hassan et al. (2011) is either highly computational with the mathematical approach or an iterative merge-split approach for collation formation. In Hadjres et al. (2018), SLA of cloud providers is given importance

for collation formation using Irvy's roommate algorithm which forms preferences for collation with not more than two cloud providers this leads to small collation size which needs next level of computation for collation formation (Irving, 1985). The previous collation formation models are either following high computational approaches or lagging in awareness of cloud providers KPI's specification which results in exhaustive search for forming collation among cloud providers. But our current approach uses concise method in solving the generation of preferences with KPI's parameters used in FLA agreement of cloud providers by using fuzzy preferences multi-decision approach for direct collation formation. Thus our scheme simultaneously analyzes SLA and FLA of cloud providers to solve the issue of collation formation and generate a differentiated model of collated providers which are ready to resolve any sort of resource provisioning issue in federated clouds.

Related Work

The concept of collation formation in the federated cloud is a result of collaboration made by cloud providers in providing resources meeting enterprise application requirements of cloud consumers. This concept offers benefits like dynamic resource provisioning, flexibility, maximized profits for different cloud providers, and improve the user experience. The purpose of creating federated clouds by cloud providers for different enterprise solutions was discussed by Rochwerger et al. (2009). In this work, the authors proposed a reservoir model which leveraged cloud providers in providing massive scale resources for meeting the infrastructure level agreements of different enterprise solutions. The proposed model highlights the need for federation for enabling cloud computing model to have features like service flexibility, controlled admission control, and optimized VM placements, cross-cloud VM allocation, monitoring, and migration. This work has not given any importance of collation formation which plays a critical role in the federation. Celesti et al. (2010) proposed cloud architecture for cross- cloud federation in which a home cloud did not fulfill the requests of its clients would forward these requests to other clouds for resolving their requirements. The cross-cloud federation formation was done by three simple steps like discovery, match making, and authentication. This work does not provide any profit maximization or sharing approach for foreign clouds. Goiri et al. (2012) proposed an economic model for federated cloud in which more public clouds were involved in federation formation given maximizing their profit. In this model, a federated provider module is developed to guide the cloud providers in taking decisions when to rent resources, get outsourced resources and turn of unused nodes. Similar to Celesti work, Giori's work does not consider profit sharing among collation formed cloud providers, nor it has given importance for heterogeneity of VM's while resource provisioning.

Different resource provisioning policies were proposed for cloud providers to maximize their profit and cloud consumers to maximize their resource utilization (Toosi et al., 2011). In this model, cloud providers were given the scope of terminating the VMs when the price for running them becomes negative. CloudSim toolkit was used to run their simulations for different workloads and check the impact of policies on effective resource management. Bittencourt and Madeira (2011) proposed a cost optimization algorithm for workflows of different applications involving high processing and storage cost on a hybrid cloud. In this work, their objective was to develop a recommended system to cloud consumers to have a decision of what amount resources to be leased from public cloud and got aggregated with a private cloud for further processing depending on his current demand. This model does not work on collation formation to maximize the profit of cloud providers and, it needs to be tested for multiple workflows.

Nordal et al. (2011) and Bin et al. (2011) proposed solutions for VM configurations in multiple clouds and cluster clouds with specific constraints like resource oriented, Performance, light-weight computations. The implementation in Nordal et al. (2011) was a new computation model named Balava to manage light-weight VM placement among multiple clouds but did not work on cloud collation formation. While the approach presented in Bin et al. (2011) highlighted on VM resilient systems to enable high availability property to perform with live migration and hardware predictive failure analysis to evacuate the VM before host system fails and provide continuous services to cloud consumers. In both approaches the cost of outsourcing was not considered. Chaisiri et al. (2012) proposed an optimal cloud provisioning algorithm to effectively manage price and demand among several cloud providers during the reservation period. In this algorithm, they used stochastic programming for maximizing profit of cloud providers. Yang et al. (2012) developed online real-time interactive applications for cloud federation architecture. This model focused on the concept of VM migration rather than resource provisioning on VM. In terms of cloud resource management, VM provisioning was proposed in several approaches. Kesavan et al. (2013) introduced a Cloud Capacity Manager to manage diverse workloads with variable demands. It failed to manage the reliability of VM for the low overhead of resources management. Rodriguez andd Buyya (2014) proposed a meta-heuristic optimization technique which aimed to minimize overall workflow execution costs within deadline requirements. This algorithm was not successful in providing elasticity and heterogeneity of computing resources. Hassan et al. (2011) proposed a horizontal dynamic cloud federation (HDCF) platform which used game theory for solving distributed resource allocation problem. Game theory cooperative and non-cooperative approaches were used to analyze the criteria of interactions among collated cloud providers while allocating resources. Mihailescu and Teo (2010) proposed a strategy-proof dynamic scheme to achieve social welfare for users while getting priced for resource usage in federated clouds. While Zhang

et al. (2015) proposed COCA, an incentive-Compatible (truthful) Online Cloud Auction mechanism to manage different user demands and generate a new bidding language to buy and sell resources. Samaan (2014) proposed an economic model to regulate the resources sharing among cloud providers for maximum profit and to meet the uncertain needs of the workload of cloud consumers. None of these approaches stated about the collation formation in the federation cloud.

Cloud federation formation was proposed in several approaches. Niyato et al. (2011) implemented a hierarchical cooperation game model which initiated resources and revenue sharing among a group of CP's by forming resources pools to avoid any uncertainty in resolving internal user's request. Mashayekhy et al. (2014) proposed a cloud collation formation approach using game theory and optimizing the maximum profit for cloud providers. In this approach, hedonic game mechanism was used to evaluate fairness properties and the collation formation is computed using split and merge algorithm. Guazzone et al. (2014) extended Mashayekhy's work by using the same hedonic gaming mechanism approach for collation formation, for an energy-aware perspective. Hassan et al. (2016) also built on Mashayekhy's work by proposing two schemes. In Hassan et al. (2016), they proposed a trust-based cooperative game model on forming collation dynamically among trustworthy collated providers to fulfill the dynamic resources request for data-intensive applications on maximizing profits and minimizing penalty cost. In Hassan et al. (2015) they focused on energy- aware federation formation using capacity-sharing mechanism and highlighted on improving the social welfare among CP's while satisfying fairness and stability properties in federation. An overlapping collation framework has been proposed to reduce the security risk involved during the federation formation (Bairagi et al., 2016). This approach depended on the cooperative game to attain higher payoffs among collated CP's. All above schemes have good technical advantages, but most of them are implemented using split and merge approach for collation formation. This mechanism has limitations like lack of stability, higher request rejection rates and failed in handling complex scenarios of VM requests because of not considering collation formation with SLA or QoS parameters which are needed measurements playing a key role while deploying in cloud production environments.

The issue of SLA-driven cloud environments was well addressed in few works like in Stanik et al. (2014), in which the authors proposed the integration for Software Defined Network (SDN) into federated cloud environments, to provide SLA guarantee for cloud consumers. The proposed approach provided an API based software components and a three-layered architectural approach using "ProgNet" to manage the SLA aware negotiation measures for federated cloud networks. In this paper, the cloud federation formation was not addressed while enforcing SLA negotiation mechanisms in cloud environments. Harsh et al. (2011) proposed another framework named contrail to support SLA and Quality

of protection agreements support for federated clouds. Contrail also created a separate layer for SLA management between cloud providers and users. The SLA requirements consideration of cloud consumers was not given importance during federation formation.

Hadjres et al. (2018) proposed an approach in which SLA requirements of cloud users were taken into consideration of collation formation. Different test scenarios were used to analyze the execution time, total profit generation, and individual payoff during collation formation. In this approach, there is a need for KPIs of each cloud provider for forming federation level agreement (FLA) which is similar to SLA not taking into consideration during collation formation. These FLA's will have much impact in contributing resources among cloud providers to meet the SLA requirements of cloud consumers. Ray et al. (2018) proposed a federation formation approach among trusted cloud providers for maximizing satisfaction level of individual cloud providers on the basis of QoS and profit using broker based cloud federation architecture. This approach failed to identify fault and QoS violation during VM migration. Agmon et al. (2018) proposed a Vickrey-Clarke-Groves (VCG) auctions for maximizing social welfare among collated cloud providers by allocating resources efficiently. Sharing of profit among collated providers is not worked out in this approach and failed in providing collision among collated cloud providers which need to be avoided. The summary of the algorithms is listed in Table 1 highlighting the facts of not choosing KPI's of CP's as major factors while forming collation which favors our study of collation formation using both SLA and FLA with KPI's for generating maximum profit among collated cloud providers.

Table 1
Summary of algorithms/model approaches used for cloud collation formation

Author Reference	Algorithm/Model Approach	Pro's	Con's
(Rochwerger et al., 2009)	Reservoir Model	It analyzed primary requirements for creation of federation in cloud computing model.	**Collation formation was not taken in to consideration with respect to SLA limitations.**
(Celesti et al., 2010)	Cross- Cloud Federation	Basic steps like discovery, match making and authentication were considered during federation with foreign clouds.	**Profit Sharing and maximization were not given importance during collation formation.**
(Bittencourt, & Madeira, 2011)	Cost Optimization Algorithm	A recommended system was built to predict the usage of resources in hybrid clouds for workflows of specific applications.	**It was not tested for multiple work flows and collation formation.**
(Nordal et al., 2011)	Balava	VM placement for multi-cloud systems was managed in this model.	**SLA Limitations were not given importance during VM placement.**

Table 1 *(continue)*

Author Reference	Algorithm/Model Approach	Pro's	Con's
(Hassan et al., 2011)	Horizontal Dynamic Cloud Federation	It solved distributed resource allocation problem during collation formation.	**Resource contentions are not dealt in meeting SLA requirements of cloud consumers.**
(Niyato et al., 2011)	Hierarchical Cooperation Game Model	This model provided solution for resource pooling and revenue sharing among cloud providers.	**SLA parameters were not taken in consideration while provisioning resources to cloud consumers.**
(Chaisiri et al., 2012)	Optimal Cloud Provisioning Algorithm	It managed price and demand of resources during reservation period.	**On demand and Spot requests of resources were not handled by cloud providers.**
(Kesavan et al., 2013)	Cloud Capacity Manager	It analyzed diverse workloads with variable demands.	**Failed in meeting reliability requirements of VM during resource management.**
(Rodriguez & Buyya, 2014)	Meta-heuristic Optimization	It handled overall workflow execution costs within deadline requirements.	**Elasticity and heterogeneity of resources were not managed.**
(Mashayekhy et al., 2014	Split and Merge Algorithm	Collation formation was achieved with fairness in profit share.	**It was exhaustive in high computational time for forming collation.**
(Hassan et al., 2015)	Energy- aware Federation Formation	Social welfare was achieved in profit sharing and stability was also attained in federation formation.	**SLA parameters were not taken into consideration during federation formation.**
(Bairagi et al., 2016)	Overlapping Collation Framework	It dealt with security risks while collation formation.	**Lead to higher user request rejection rate.**
(Hadjres et al., 2018)	SLA-Aware Collation Algorithm	SLA parameters with Irvy roommate algorithm was used to improve the collation formation.	**High computational time in finding preferences and forming collation and KPI factors of CP's were not considered during collation formation.**
(Ray et al., 2018)	Broker Based Cloud Federation Architecture	It dealt with satisfaction of individual cloud providers in maximizing profits during collation formation	**SLA and KPI's were not dealt in this architecture model.**
(Agmon et al., 2018)	Vickrey-Clarke-Groves (VCG) Auctions Model	It analyzed profit sharing effectively using social welfare.	**Collision of CP's while sharing resources within SLA limitations was overlooked during collation formation**

METHODS

An FLA-SLA Aware Cloud Coalition Formation Framework

This section presents the FLA-SLA Aware Cloud Coalition Formation (FS-ACCF) problem statement, details of the layered federated cloud model and architecture diagram of FS-ACCF model with mathematical formulation and describes the coalition formation algorithm.

Layered Federated Cloud Model

Figure 1 gives the layered federated cloud model which was considered to simulate the understanding of federated cloud approach. It consisted of the provider agent layer, Collated provider agent layer, Broker agent layer, Consumer agent layer. Each layer had set of agents which were involved in interaction to provide the idea of federation formation in cloud computing. The cloud consumer agents would request for resources with desired SLA requirements. In this model, the request of resources was done by considering the number of VMs of different types like small VMs, Medium VMs, Large VMs, and Extra large VMs. Table 2 shows VM instance characteristics of each VM type and these characteristics, prices were inspired by On-Demand instances of Amazon EC2. These requests were forwarded to cloud provider agents through broker agents and then collation formation process was started by having interaction among the set of provider agents and set of collated provider agent's layer was formed for serving the consumer request for a particular instance.

In this model set of KPI factors like (Uptime, Downtime, Reqtimein. Reqtimeout, Inbyte, Outbyte, Packsize, Availbandwidthin, Availbandwidthout, Packtimein, Packtimeout, Disksize) of cloud provider agents were considered for collation formation. The KPI's were provided by each cloud provider and FLA were computed based on these values.

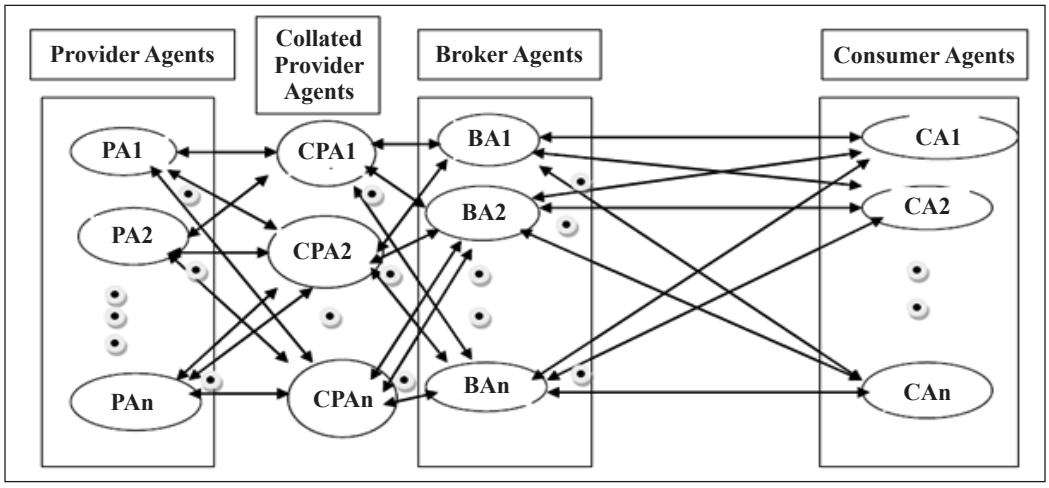

Figure 1. Layered federated cloud model

Architecture Diagram for FLA-SLA Aware Cloud Collation Formation Model

Figure 2 is a proposed architecture model in the above layered federated cloud model to analyze different KPI factors for FLA generation among different cloud providers and compute required SLA parameter values at various instances as the request made by cloud consumers. The cloud broker agent's forwards the request made by cloud consumers along with required SLA parameters. In this architecture model the user required SLA parameters were analyzed over cloud providers computed FLA values along with KPI factor.

Table 2
Example of VM configurations

Parameters	Small VM	Medium VM	Large VM	Extra Large VM
Number of Cores (1.6 GHz CPU)	1	2	4	8
Memory (GB)	1.7	3.75	7.5	15
Storage (TB)	22	48	98	199
Price	0.12	0.24	0.48	0.96

A set of collated provider's satisfying these consumers request at that instance were send for computing Fuzzy Relationship using multi-criteria decision approach for simultaneously analyzing collated providers collation formation for different SLA parameters. In this paper the different SLA Parameters were treated as X Parameters for FS-ACCF algorithm, where the X was used for Availability, Response time, and Process time.

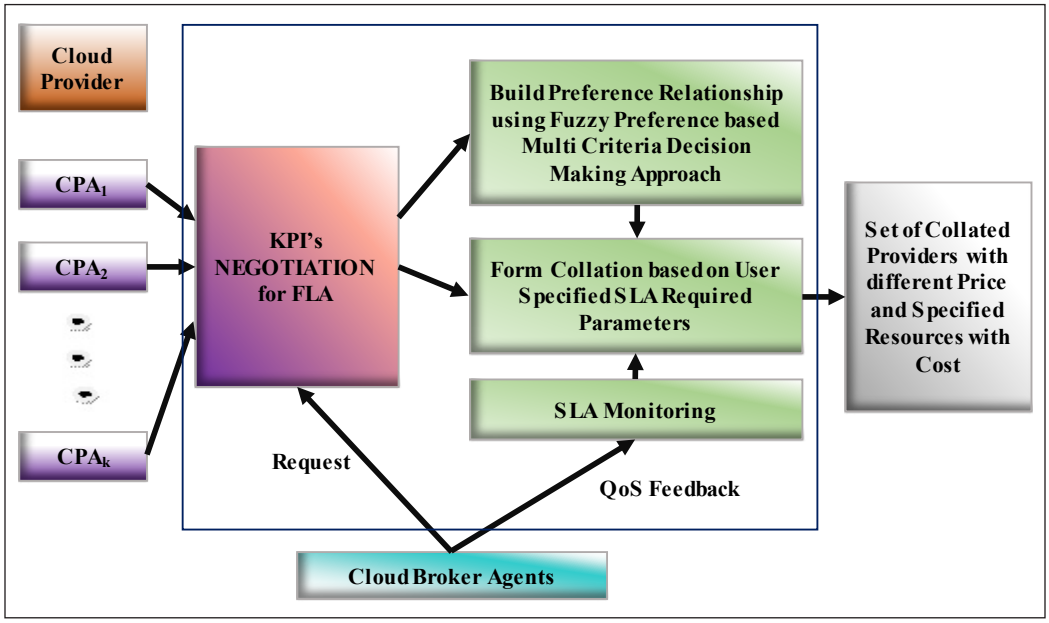

Figure 2. Architecture for FLA-SLA aware cloud collation formation model

Fuzzy preference relationship provides required collated providers list along with their cost at which they provide resources for cloud consumers. The list of collated providers was analyzed at different instances for providing resources. The cloud broker job was to provide this list of collated providers and their cost information to cloud consumers so that they can have a choice of service classes for their different SLA parameters request of resources. Fuzzy preference relationship provides required collated providers list along with their cost at which they provide resources for cloud consumers. The list of collated providers was analyzed at different instances for providing resources. The cloud broker job was to provide this list of collated providers and their cost information to cloud consumers so that they can have a choice of service classes for their different SLA parameters request of resources.

Notations. The set of Cloud providers specified by CP= {I/I=1, 2….N} and User request vector RV_k= {rv_1, rv_2….rv_m} rest of parameters and their descriptions are specified in Table 3 notations.

The best collation of provider agents that satisfied a user request along with their specified SLA parameters was found. The complex task was to analyze the KPIs of cloud

Table 3
Notations

Variables	Description
N	No of available CPs
CP	A set of Cloud Providers CP= {I/I=1, 2, … N}
M	Number of types of VMs i.e. smallVM, mediumVM, LargeVM, ExtralargeVM for example (5,10,6,4)
VM_{ij}	VM of size j offered by provider i
RV_k	User Request Vector
$Cost_i$	A set of cost of VM's provided by i provider
rv_{kj}	No of VM 's of type j needed by user request
$AvailP_i$	Availability FLA parameter computed by a provider i using specified KPIs
PtP_i	Process time FLA parameter computed by a provider i using specified KPIs
RtP_i	Response time FLA parameter computed by a provider i using specified KPIs
X_{ij}	Decision variable that represents the no of VM's of allocated by provider i
F	Characteristic function that is used to obtain Total profit obtained by collation
ColP	The set of all collations that can be formed from CPSs.
TotalProfitX	Total Profit earned by different collation formed on X parameters. Where X is Availability, Response time and Process time.
ShapelyValueX	Shapley Value computed for each cloud provider involved in collation formation on X parameters Where X is Availability, Response time and Process time
FXPrice	A set of prices proposed by the broker for each type of VM FXPrice= $price_1$, $price_2$, … $price_m$}

providers for computing FLA parameters and then the preferred combinations of cloud providers who satisfied the user request with required SLA parameters were found. In this system model the preference of cloud providers were computed using fuzzy preference relationship multi-decision approach for collation formation among cloud providers. Total profit was computed based on the price set by broker and cost computed by their preferred collation cloud providers to check for maximum profit earned by the collation formation. This maximum profit needed to be shared among collated providers based on their contribution of resources.

System Model and Mathematical Formulation

We modeled the FS-ACCF problem as a hedonic coalitional game (CP, V) with transferable utility (*TU*). A cooperative game as mentioned in Saad et al. (2009) and Álvarez-Mozos et al. (2013) is a set of players will cooperate to form collations and share profit among them based on their contribution in collation. A utility or characteristic function is used to measure the total profit of the possible collation by meeting specific requirements and condition in collation formation. In this system model the preference relation is established are computed based on the fuzzy preference relationship of multi-decision approach.

Collation Formation Model

In our case of hedonic game, the players were cloud providers among which their FLA parameters were computed using their specified KPIs and then performed fuzzy preference relationship to list the collated cloud providers who satisfied the cloud consumer request at that instance. In our system the SLA parameters between the cloud providers was considered as FLA and the values were computed by specified KPI's (i.e. Uptime, Downtime, Reqtimein. Reqtimeout, Inbyte, Outbyte, Packsize, Availbandwidthin, Availbandwidthout, Packtimein, Packtimeout, Disksize) of cloud providers by Equation [1], [2], [3], [4] and [5].

$$AvailP_i = 1 - \left(\frac{Downtime_i}{Uptime_i} \right), \text{ where i} \epsilon \text{ CP} \qquad [1]$$

$$PtP_i = \text{Re}\, qtimeout_i - \text{Re}\, qtimein_i, \text{ where i} \epsilon \text{ CP} \qquad [2]$$

$$RtP_i = RinP_i + RoutP_i, \text{ where i} \epsilon \text{ CP} \qquad [3]$$

$$RinP_i = \frac{Packsize_i}{\left(Availbandwidthin_i - Inbyte_i \right)} \qquad [4]$$

$$RoutP_i = \frac{Packsize_i}{\left(Availbandwidthout_i - Outbyte_i \right)} \qquad [5]$$

Cloud provider computed their respective FLA parameters by above equations then each value of this compared with SLA parameter mentioned along with user request vector (RV_k). Initially the cloud providers whose FLA parameters not matching with the SLA values were eliminated from collation formation for that instance. The rest of cloud providers were considered for fuzzy preference relation multi-decision approach for checking preference relationship and forming collation preferences with respect to that FLA parameter values. The process of this fuzzy preference relation multi-decision approach was taken from Chang and Wang (2008), Tanino (1988), Hipel et al. (2011) and Mesiar (2007). In these approaches they had been applied this fuzzy preference computation for gathering preferences in different computation fields like WIMAX and System design decision.

The final lists of preferred collated providers were used for total profit computation. A characteristic function (i.e. F) was used to associate profit to a collation. Thus, total profit was a real valued function F: ColP $\rightarrow \mathbb{R}^+$ where F ($\emptyset$) =0 and $\emptyset$ is empty collation. A provider would get many choices of collated providers list generated because of fuzzy preferences relationship multi-decision approach. Now providers should choose which collation providers list would generate maximum total profit.

The best collation of cloud providers was analyzed by satisfying all requirements while maximizing total profit. This can be expressed as follows:

$$Max \sum_{CP_i \in C} \sum_{j=1}^{M} X_{ij}(Fprice_i - \mathrm{Cos}\, t_{ij} - \alpha_i PenaltyFunc(X)) \qquad [6]$$

$$\text{where } PenaltyFunc(X) = \sum_{j=1}^{n} \frac{\sum_{k=1,k\neq P}^{|C|} X_{kj} \mathrm{Cos}\, t_{kj}}{X_{ij} \mathrm{Cos}\, t_{ij}} \qquad [7]$$

Subject to:

N, M $\in$ N (Cond1)

F $\in \mathbb{R}^+$ (Cond2)

$X_{ij} \geq 0$ for all i=1…N and j=1..4 (Cond3)

$\sum_{i}^{j} X_{ij} = r_{kj}$ for j=1..4 (Cond4)

$X_{ij} \leq VM_{ij}$ for i=1..n and j=1..4 (Cond5)

α_i= { 0,1} (Cond6)

The Equation (6) was used for computing maximum total profit for selected decision variable instances with fixed price; cost for required VM's allocated with penalty function. The penalty function of Equation (7) was applied by cloud providers CP_i to compute penalty cost for not supplying the resources with specified SLA values after collation formation. It was instantaneously computed for cost associated with those VMs of cloud providers which were not meeting their specified contribution among collated providers. Constraints (Cond1) and (Cond2) ensured the number of VMs and number of cloud providers to be set of Natural numbers and real positive numbers respectively. Constraints (Cond3) for the decision variable X_{ij} was a Positive Value. The constraint (Cond4) gives the request of that particular VM instance is meeting with their summated decision variable of all collated providers involved in collation formation. The constraint (Cond5) ensured that decision variable was not exceeding required VM instances. The final constraint (Cond6) was the decision parameter for penalty function existing or not while provisioning resources after forming collation.

Profit Sharing Model

Once the best collation formed, the total profit was obtained for particular instance of user request with specified SLA parameter. The total profit was distributed among those cloud providers who were contributing to the collation done based on normalized Shapley value computation. The Shapley value payoff was obtained by the product of collation profit by the normalized Shapley Value:

$$ShapleyPayoffX_i = \frac{ShapleyValueX_i}{\sum_{CP_j \in C} ShapleyValueX_i} * \Pr ofit \quad [8]$$

$$ShapleyValueX_i = \sum_{subC \subset comb(C \setminus \{CP_i\}} \frac{(S!(n_C - S - 1)!}{n_C!}(v(subC \cup \{CP_i\}) - v(subC)) \quad [9]$$

Where S is cardinal of the sub collation subC,n_c is the cardinal of C and Comb(C) to the set of all combinations of 1,2,….., n_c elements of C.

The Equation [8] was used for computing shapley payoff value for on a particular QoS parameter by normalizing shapley value for sharing profit among the collated cloud providers. The Equation [9] computed the Shapley value on each QoS parameter for sub-collation and collation set of cloud providers for sharing profit. The Figure 3 gives clear summary of integration of above both models.

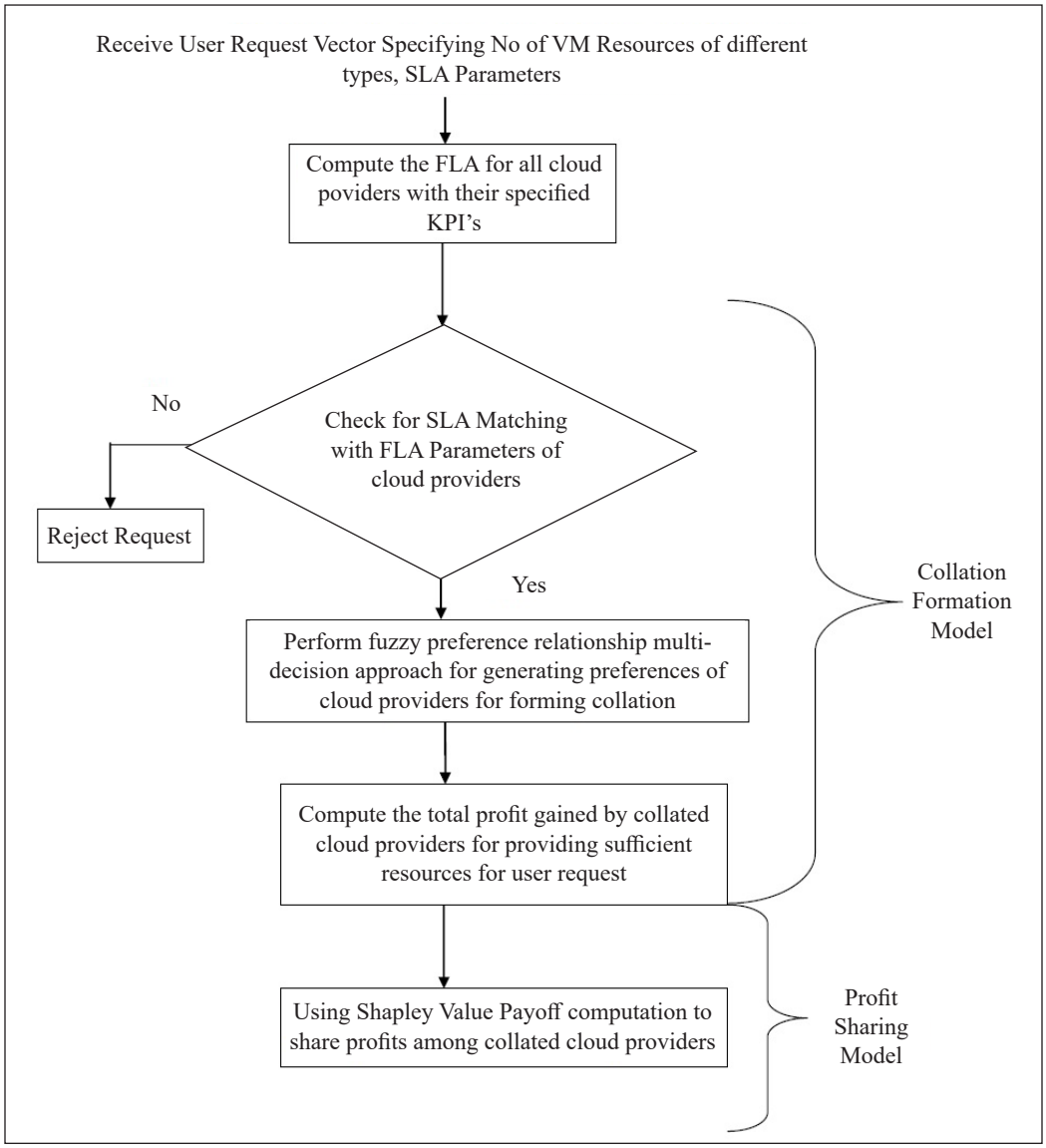

Figure 3. Dataflow diagram for FS-ACCF

FLA-SLA Aware Collation Formation Algorithm

Algorithm 1: FLA-SLA Aware Collation Formation Algorithm (FS_ACCF)

1. **Input:**the cloud providers set CP and their KPI's(Key Performance Indicator's the request R,SLA parameters
2. Calculate the FLA parameters from the specified KPI's of cloud providers
3. Eliminate from Set of CP all CP's that do not meet the required SLA parameters

4. XCollatedPreferences=FuzzyPreferencesListX(NewCPlist, XparametersofNewCP); (X will be Availability, Response time, Process time etc..)/* Calculating Fuzzy Preference relation using multi criteria decision making approach */
5. [TotalProfitX]=Find_CollationX(XCollatedPreferences,UserReq,CPRes,Cost);/* Calculated total profit earned by the collated providers during collation and providing resources without violation of SLA parameters.
6. [ShapelyValueX]=CalShapelyValueX(XCollatedPreferneces,TotalProfitX, CPRes,Cost);/*calculate Share of profit among the collated provider using shapely value */
7. Validate the resource availability during collated providers providing resources.
8. If Current selected XCollatedPreferences doesn't provide enough resources, then UserReq will be checked for other XCollatedPreferences and get allocated with different price.

The Algorithm 1 was used for processing the request made by cloud consumers along with their specified SLA parameters. Initially in step 2 of algorithm the set of KPI's were used for undertaking FLA agreement among cloud providers. In step 3 SLA parameters of cloud consumers were checked for FLA agreed values of cloud providers and selecting only those cloud providers matching with in SLA parameters limit. The step 4 is called the fuzzy preferences list Algorithm 2 by passing selected providers list with required X SLA parameters of those providers.

The Step 7 was used for calculating Total profit for those X SLA parameters with preferred Collated providers by checking their satisfaction of cloud consumer's request of resources within their SLA parameter limit. Step 9 was used for Shapley value calculation to share the profit among the collated providers. Step 11 and 12 were used if any of the selected collated providers fail to meet in providing required resources within their SLA limit.

Algorithm 2: Computing the Collated Preferences lists

Function FuzzyPreferencesListX (NewCPlist, Xparameter)

1. Count = Length(NewCPlist)
2. For i=1:Count assign array A(i)= Xparameter(NewCPlist(i); end for
3. For i=1:Count For j=1:Count Intialize 2-dimensional matrix CurrA for FAHP computation.
4. If(i==j) CurrA(i,j)=1 else compute m=A(i)-A(j) end if
   If(m <0) CurrA(i,j)=0 else CurrA(i,j)=m end if end for end for

5. For i=1:Count For j=1:Count fuzzyX(i,j)=max((currA(i,j)-currA(j,i)),0); end for end for
6. For i=1:Count fuzzymaxX(i)=1-max(fuzzyX(:,i)); end
7. For i=1:Count For j=1:5 if(fuzzymaxX (i)==Xparameters (j)) finallist(i)=j; end if end for end for
8. for i=1:Count For j=1:Count if(fuzzymaxX(i)<=fuzzymaxX(j)) preferlistX(i,j)=finallist(j); end if end for end for
9. XCollatedpreferences=preferlistX;
10. Return (XCollatedPrefernces)

The Algorithm 2 Fuzzy Preferneces List is purely a mathematical model multivariable decision approach used for calculating preferences among the cloud providers with their agreed FLA values of different X SLA parameters. It provides output as different collated providers list. At step 2 one dimensional matrix A(I) is initialized with new cloud providers SLA parameter values. At step 3 and 4 a two dimensional matrix CurrA (I,J) was initialized by comparing the A(i) values. At step 5 fuzzyX matrix was computed to get the maximun value of CurrA (I,J).At step 7 and 8 the finalist of preferred cloud providers for forming collation were listed by undergoing the fuzzy computation.

The Algorithm 3 was used for computing the total profit for different X Collated preferences simultaneously by collated cloud providers satisfying user requested and checked for their availability of resources and calculateed the price which needed to be paid by cloud consumers for getting serviced by that collated providers.

Algorithm 3: Computing the total profit of XCollatedPreferences

Function Find_CollationX(XCollatedPreferences,UserReq,CPRes,Cost)

1. For each XCollatedPreferences list try to check for availability of resources and cost satisfy the UserReq and CPRes
2. CPRes(XCollatedPreferences)>UserReq(Resources)
3. profiteachXCollated=Cost(XCollatedPreferences)*CPRes(XCollatedPreferences)
4. UserReq(Resources)=UserReq(Resources)-CPRes(XCollatedPredferences)
5. And repeat above steps for each XCollatedPreferences until UserReq(Resources) are satisfied.
6. TotalProfitX=addall(profiteachXCollated)
7. Return(TotalProfitX)

Algorithm 4: Computing the ShapleyVal of XCollatedPreferences

Function CalShapelyValueX(XCollatedPreferences ,TotalProfitX,Cost,CPRes)

1. For each XCollatedPreferences TotalProfitX calculate
2. ShapleyValX=Factorial(n-1)*Factorial((n-(n-1)-1)/Factorial(n) * (TotalProfitX)-Cost(XCollatedPreferences)
3. List all ShapleyValX of each XCollatedPreferences

The Algorithm 4 was used for Shapley value calculation to share the total profit among the specified collated providers who collaboratively satisfied the consumer's request within his specified SLA limit.

RESULTS AND DISCUSSIONS

We implemented the FS-ACCF algorithm using MATLAB. The formulated computations were done through MATLAB code. The results were obtained by executing the FS-ACCF algorithm and compared with S-ACCF algorithm. The S-ACCF algorithm considered Irvy's roommate algorithm to pair the collated providers with minimum combination to achieve the collation with few SLA parameters and to attain total profit computations. But in the collation formation in FS-ACCF algorithm the KPI factors of cloud providers could be used for forming collation using fuzzy set approach of multi-decision criteria for different FLA parameters at particular instance and total profit are computed.

Simulation Environment and Evaluation Metrics

The simulation environment was carried through MATLAB for 6 to 10 consecutive requests for particular VM configurations as mentioned in Table 2, Table 4 gives the cloud providers with their SLA parameters and KPI's and Table 5 gives the resulted values of execution time and total profit during collation formation by both FS-ACCF and S-ACCF algorithms at different instances and tested for collation formation on Intel corei3 processor, 4GB RAM with Windows 7.

Evaluation Metrics. Five(5) metrics were used to evaluate the two approaches: 1) The execution time of the collation formation algorithms for different requests at different instances; 2) The total profit generated by collation; 3) The individual payoff for each provider in the collation for specific XFLA Parameter where X i.e. availability, response time and process time; 4) The no of providers in the generated collation; 5) The number of VMs per provider in the collation.

Performance Results and Analysis

The Figure 4 shows the execution time for different user requests and the comparsion between S-ACCF algorithm and FS-ACCF algorithm for execution time with specific SLA parameters request made by them. It is clear from this graph that FS-ACCF using Fuzzy Preference Multi-decision appraoch resolved the collation formation much faster than S-ACCF Irvis rommmate algorithm. In Figure 4, the maximum difference between the execution time is because in S-ACCF algorithm the collation formation is checked only between pair of two cloud providers which take more time for forming collation where as in FS-ACCF algorithm many pair of collated cloud providers where simultaneously many QoS parameters are considerd for collation formation with less execution time.

Table 4
List of SLA and KPI'S values of cloud providers participated in collation formation

Cloud Providers	SLA Parameters (Availability, Response Time ,Process Time)	KPI'S (Uptime, Downtime, Reqtimein. Reqtimeout, Inbyte, Outbyte, Packsize, Availbandwidthin, Availbandwidthout, Packtimein, Packtimeout, Disksize)
CP1	(0.989,0.36,0.23)	**(5,10,0.253,0.415,5,6,1024,10,20,0.67,0.68,2)**
CP2	(0.956,0.55,0.20)	**(7,12,0.415,0.283,8,10,2048,23,45,0.54,0.87,4)**
CP3	(0.975,0.43,0.26)	**(2,8,0.645,0.923,3,4,512,21,56,0.23,0.45,1)**
CP4	(0.968,0.67,0.45)	**(4,10,0.93,0.283,4,7,4096,112,34,0.65,0.78,8)**
CP5	(0.945,0.56,0.34)	**(12,6,0.676,0.459,5,7,256,23,45,0.85,0.56,3)**

Table 5
List of cloud consumers request vectors with resulted execution time and total profit gained during collation formation using FS-ACCF Algorithm and S-ACCF Algorithm

S.NO	Cloud Consumer Request Vector for Resources i.e No of (SmallVM, MediumVM, LargeVM, ExtralargeVM)	Execution Time taken for Collation Formation using FS-ACCF Algorithm	Execution Time taken for Collation Formation using S-ACCF Algorithm	Total Profit Gained by Forming Collation Using FS-ACCF Algorithm	Total Profit Gained by Forming Collation Using S-ACCF Algorithm
1	(3,5,8,9)	0.0014	0.0143	144.5	**137.9**
2	(13,15,18,19)	0.0029	0.0101	93.8	**69.1**
3	(10,5,18,19)	0.0042	0.0093	307.4	**267.9**
4	(20,5,18,19)	0.0015	0.0105	272.4	**207.9**
5	(2,6,8,9)	0.0022	0.011	310.4	**198.6**
6	(12,16,6,9)	0.0013	0.0088	92.7	**69.23**

Figure 5 shows the comparison of total profit for different user request specified response time SLA parameter between FS-ACCF algorithm and S-ACCF algorithm. The S-ACCF algorithm only considered one SLA parameter that is response time but in FS-

ACCF multiple SLA parameters are considered and analyzed simultaneously to check for resources by providing optimal choice for cloud consumers in selecting the collated providers within their price limit.

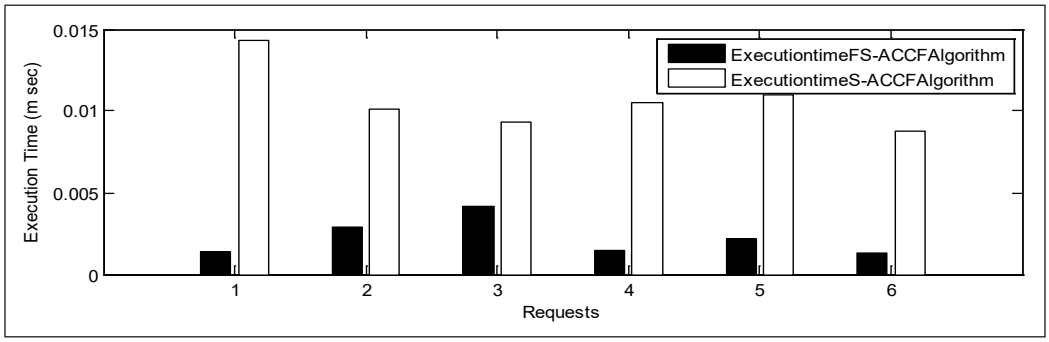

Figure 4. Execution time vs requests for FS-ACCF and S-ACCF algorithms

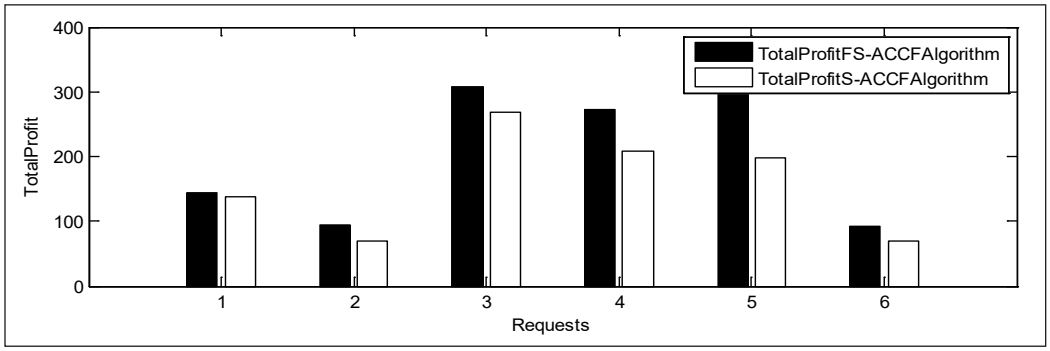

Figure 5. Total Profit vs requests of FS-ACCF and S-ACCF algorithms

Figure 6 gives the total profit for different XFLA parameters using FS-ACCF algorithm, here the X is Availability, Response time and Process Time which are the FLA parameters calculated based on FLA agreement among collated cloud providers with their specified KPI's.In the above graph X-axis Totalprofit(;,1)1 is for total profit computed for Availability FLA Parameter Collated providers, Totalprofit(:,2)2 for total profit computed for Response time FLA Parameter Collated providers and Totalprofit(:,3)3 for total profit computed for Process time FLA Parameter Collated providers.

Figure 7 depicts the no of VM's(NoofVMs(:,1) Small,NoofVMs(:,2) Medium,NoofVMs(:,3) Large,NoofVMs(:,4)ExtaLarge) participated in collation formation for different FLA parameters on X-axis 1 for Availability, 2 for Response time ,3 for Process time using FS-ACCF algorithm. These are the results generated after running the fuzzy set approach of multi-decision criteria for different instance requests made by user for specified SLA parameter.

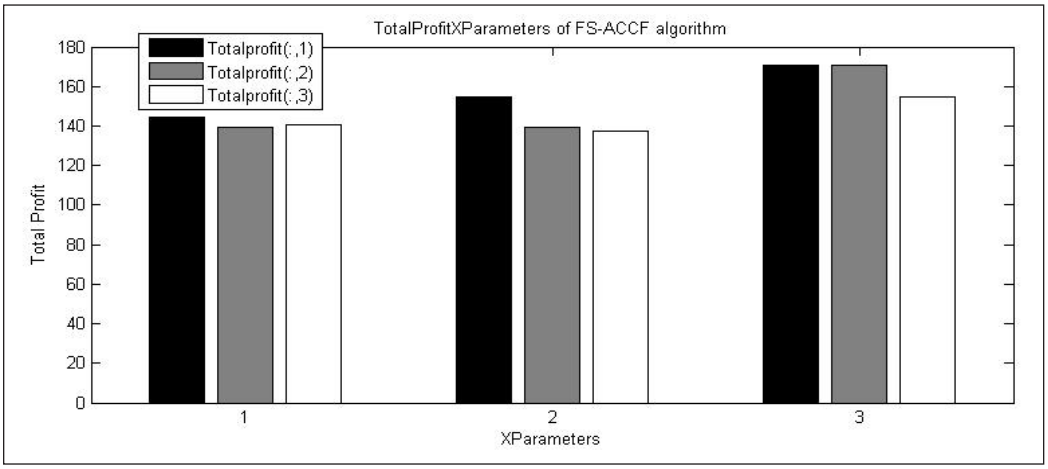

Figure 6. Total profit for different X parameters using FS-ACCF algorithm

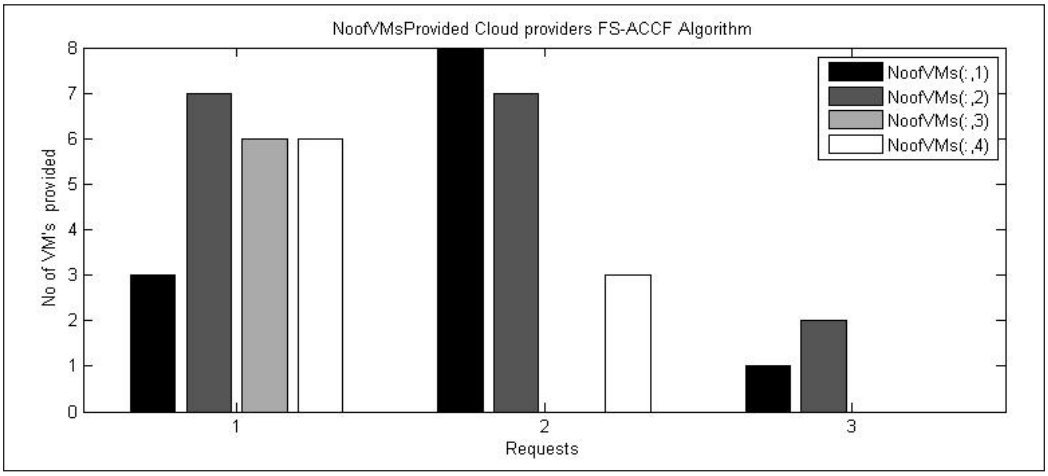

Figure 7. No of VM's participated in collation using FS-ACCF algorithm

Figure 8 gives the Shapley value for different collated providers sharing profit after forming collation on specific FLA parameter.The ShapleyValue(:,1) is for Shapley value computation for availability ,ShapleyValue(:,2) is for shapley Value computation for response time ,ShapleyValue(:,3) for shapley value computation for process time by using FS-ACFF algorithm.

The Figure 8 only provides the sharing of total profit among the cloud providers who are involved in collation. The Figure 6 justifies the main objective of our paper which results in maximizing total profit for different SLA parameters and Figure 5 gives comparison of total profit generated for different consecutive requests for both FS-ACCF and S-ACCF algorithms.

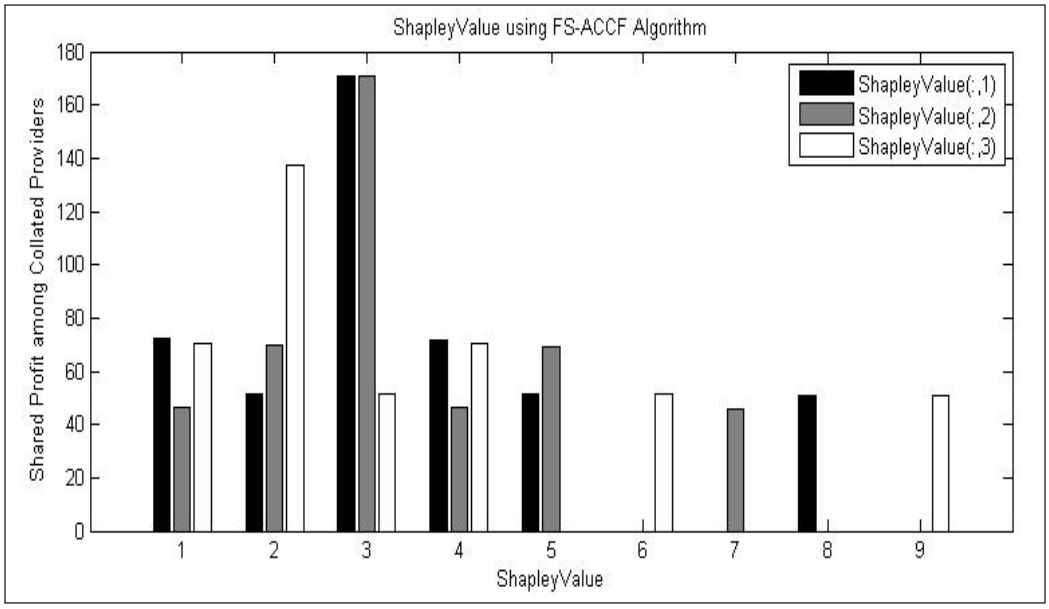

Figure 8. Shapley value of different XCollated providers using FS-ACCF algorithm

CONCLUSION

In this paper different flavor collation formation approach i.e. FS-ACCF algorithm was compared with S-ACCF algorithm in terms of total profit and sharing of profit among collated providers. The basic difference is in terms of generating preferences list from both approaches. In S-ACCF approach they used Irv's roommate algorithm for generating collation preferences which was having less time complexity in computation. Our approach is using fuzzy set multi-decision approach for preference relationship computation along with list of collated providers from whom the user's will have a perfect choice to choose a collated providers list who provides resources with maximum profit. Our current FLA-SLA collation formation was done on computing resources KPI factors for computing FLA values among cloud providers. The future work is to consider the storage KPI factors for computing FLA parameters and broadly try for all possible combinations of FLA parameters during collation formation for generating maximum profit and meeting all SLA parameter requirements of cloud consumers.

ACKNOWLEDGEMENT

We thank Dr. B. Thirumala Rao, Professor, K L University who provided insight and expertise that greatly assisted the research, although he might not agree with all the interpretations/conclusions of this paper. We also would like to show our gratitude to the Dr. Raj Kumar Buyya, Professor, University of Melbourne, for sharing his pearls of wisdom

with us during the course of this research, and we thank 3 "anonymous" reviewers for their so-called insights. We are also immensely grateful for their comments on an earlier version of the manuscript, although any errors are our own and should not tarnish the reputations of these esteemed persons.

REFERENCES

Agmon, S., Ben-Yehuda, O. A., & Schuster, A. (2018). Preventing collusion in cloud computing auctions. In *International Conference on the Economics of Grids, Clouds, Systems, and Services* (pp. 24-38). Cham, Switzerland: Springer.

Álvarez-Mozos, M., van den Brink, R., van der Laan, G., & Tejada, O. (2013). Share functions for cooperative games with levels structure of cooperation. *European Journal of Operational Research*, *224*(1), 167-179.

Bairagi, A. K., Alam, M. G. R., Talukder, A., Nguyen, T. H., & Hong, C. S. (2016, January 13-15). An overlapping coalition formation approach to maximize payoffs in cloud computing environment. In *2016 International Conference on Information Networking (ICOIN)* (pp. 324-329). Kota Kinabalu, Malaysia.

Bin, E., Biran, O., Boni, O., Hadad, E., Kolodner, E. K., Moatti, Y., & Lorenz, D. H. (2011, June 20-24). Guaranteeing high availability goals for virtual machine placement. In *2011 31st International Conference on Distributed Computing Systems* (pp. 700-709). Minneapolis, MN, USA.

Bittencourt, L. F., & Madeira, E. R. M. (2011). HCOC: a cost optimization algorithm for work flow scheduling in hybrid clouds. *Journal of Internet Services and Applications*, *2*(3), 207-227.

Celesti, A., Tusa, F., Villari, M., & Puliafito, A. (2010, July 5-10). How to enhance cloud architectures to enable cross-federation. In *2010 IEEE 3rd International Conference on Cloud Computing* (pp. 337-345). Miami, FL, USA.

Chaisiri, S., Lee, B. S., & Niyato, D. (2012). Optimization of resource provisioning cost in cloud computing. *IEEE Transactions on Services Computing*, *5*(2), 164-177.

Chang, T. H., & Wang, T. C. (2008, June 1-6). Fuzzy preference relation based multi-criteria decision making approach for WiMAX license award. In *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)* (pp.37-42). Hong Kong, China.

Chen, J., Wang, C., Zhou, B. B., Sun, L., Lee, Y. C., & Zomaya, A. Y. (2011, June 8-11). Tradeoffs between profit and customer satisfaction for service provisioning in the cloud. In *Proceedings of the 20th international symposium on High performance distributed computing* (pp. 229-238). San Jose, California, USA.

Goiri, Í., Guitart, J., & Torres, J. (2012). Economic model of a cloud provider operating in a federated cloud. *Information Systems Frontiers*, *14*(4), 827-843.

Guazzone, M., Anglano, C., & Sereno, M. (2014, May 26-29). A game-theoretic approach to coalition formation in green cloud federations. In *2014 14th IEEE/ ACM International Symposium on Cluster, Cloud and Grid Computing* (pp. 618-625). Chicago, IL, USA.

Hadjres, S., Kara, N., El Barachi, M., & Belqasmi, F. (In Press). An SLA-aware cloud coalition formation approach for virtualized networks. *IEEE Transactions on Cloud Computing*. doi: 10.1109/ TCC.2018.2865737

Harsh, P., Jegou, Y., Cascella, R. G., & Morin, C. (2011). Contrail virtual execution platform challenges in being part of a cloud federation. In W. Abramowicz, I. M. Llorente, M. Surridge, A. Zisman & J. Vayssière (Eds.), *European Conference on a Service-Based Internet* (pp. 50-61). Heidelberg, Germany: Springer.

Hassan, M. M., Al-Wadud, M. A., Almogren, A., Song, B., & Alamri, A. (2015). Energy-aware resource and revenue management in federated cloud: a game-theoretic approach. *IEEE Systems Journal*, *11*(2), 951-961.

Hassan, M. M., Al-Wadud, M. A., Almogren, A., Rahman, S. M. M., Alelaiwi, A., Alamri, A., & Hamid, M. A. (2016). QoS and trust-aware coalition formation game in data-intensive cloud federations. *Concurrency and Computation: Practice and Experience*, *28*(10), 2889-2905.

Hassan, M. M., Song, B., & Huh, E. N. (2011, September 2-4). Distributed resource allocation games in horizontal dynamic cloud federation platform. In *2011 IEEE International Conference on High Performance Computing and Communications* (pp. 822-827). Banff, AB, Canada.

Hipel, K. W., Kilgour, D. M., & Bashar, M. A. (2011). Fuzzy preferences in multiple participant decision making. *Scientia Iranica*, *18*(3), 627-638.

Irving, R. W. (1985). An efficient algorithm for the "stable roommates" problem. *Journal of Algorithms*, *6*(4), 577-595.

Kesavan, M., Ahmad, I., Krieger, O., Soundararajan, R., Gavrilovska, A., & Schwan, K. (2013). Practical compute capacity management for virtualized datacenters. *IEEE Transactions on Cloud Computing*, *1*(1), 1-1.

Mashayekhy, L., Nejad, M. M., & Grosu, D. (2014). Cloud federations in the sky: Formation game and mechanism. *IEEE Transactions on Cloud Computing*, *3*(1), 14-27.

Mesiar, R. (2007). Fuzzy set approach to the utility, preference relations, and aggregation operators. *European Journal of Operational Research*, *176*(1), 414-422.

Mihailescu, M., & Teo, Y. M. (2010, May 17-20). Dynamic resource pricing on federated clouds. In *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing* (pp. 513-517). Melbourne, VIC, Australia.

Mishra, A., Jain, R., & Durresi, A. (2012). Cloud computing: Networking and communication challenges. *IEEE Communications Magazine*, *50*(9), 24-25.

Niyato, D., Vasilakos, A. V., & Kun, Z. (2011, May 23-26). Resource and revenue sharing with coalition formation of cloud providers: Game theoretic approach. In *2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (pp. 215-224). Newport Beach, CA, USA.

Nordal, A., Kvalnes, Å., Hurley, J., & Johansen, D. (2011, July 4-9). Balava: Federating private and public clouds. In *2011 IEEE World Congress on Services* (pp. 569-577). Washington, DC, USA.

Ray, B., Saha, A., Khatua, S., & Roy, S. (2018, In Press). Quality and profit assured trusted cloud federation formation: game theory based approach. *IEEE Transactions on Services Computing.* doi: 10.1109/TSC.2018.2833854

Rochwerger, B., Breitgand, D., Epstein, A., Hadas, D., Loy, I., Nagin, K., & Levy, E. (2011). Reservoir-when one cloud is not enough. *Computer*, *44*(3), 44-51.

Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., ... & Ben-Yehuda, M. (2009). The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, *53*(4), 4-11.

Rodriguez, M. A., & Buyya, R. (2014). Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. *IEEE Transactions on Cloud Computing*, *2*(2), 222-235.

Saad, W., Han, Z., Debbah, M., Hjørungnes, A., & Basar, T. (2009). Coalitional game theory for communication networks: A tutorial. *IEEE Signal Processing Magazine, 26*(5), 77-97.

Samaan, N. (2014). A novel economic sharing model in a federation of selfish cloud providers. *IEEE Transactions on Parallel and Distributed Systems*, *25*(1), 12-21.

Stanik, A., Koerner, M., & Lymberopoulos, L. (2014). SLA-driven federated cloud networking: Quality of service for cloud-based software defined networks. *Procedia Computer Science*, *34*, 655-660.

Tanino, T. (1988). Fuzzy preference relations in group decision making. In J. Kacprzyk & M. Roubens (Eds.), *Non-conventional preference relations in decision making* (pp. 54-71). Heidelberg, Germany: Springer.

Toosi, A. N., Calheiros, R. N., Thulasiram, R. K., & Buyya, R. (2011, September 2-4). Resource provisioning policies to increase iaas provider's profit in a federated cloud environment. In *2011 IEEE International Conference on High Performance Computing and Communications* (pp. 279-287). Banff, AB, Canada.

Yang, X., Nasser, B., Surridge, M., & Middleton, S. (2012). A business-oriented cloud federation model for real-time applications. *Future Generation Computer Systems*, *28*(8), 1158-1167.

Zhang, H., Jiang, H., Li, B., Liu, F., Vasilakos, A. V., & Liu, J. (2015). A framework for truthful online auctions in cloud computing with heterogeneous user demands. *IEEE Transactions on Computers*, *65*(3), 805-818.