A Multi-Cloud Marketplace Model with Multiple Brokers for IaaS Layer and Generalized Stable Matching

Shrenik Jain
Computer Systems Group
IIIT Hyderabad
shrenik.jain@research.iiit.ac.in

Suresh Purini
Computer Systems Group
IIIT Hyderabad
suresh.purini@research.iiit.ac.in

Puduru V. Reddy
Department of Electrical Engineering
Indian Institute of Technology
Chennai, India
vishwa428@gmail.com

***Abstract*—In this paper, we propose a multi-cloud marketplace model for Infrastructure-as-a-Service (IaaS) layer with multiple cloud providers, intermediate brokers and end users. The brokers service end users subscribed to them by aggregating resources (virtual machines) from cloud providers while maximizing their profits. Similarly cloud providers (producers) allocate their supply of virtual machines to brokers (consumers) so as to maximize their profits. We define the notion of social welfare in this market structure and study two trading schemes. The first scheme involves centralized control which aims at maximizing social welfare but may contain unstable producer-consumer pairs who have an incentive to deviate from the current allocation. The second scheme eliminates such unstable pairs by using a generalization of stable matching algorithm but may lead to sub-optimal social welfare. The stable matching algorithm we proposed in this paper is a particular way of generalizing the original Gale-Shapley algorithm.**

***Keywords*-Inter-Cloud, Multi-Cloud, Federated Clouds, Brokerage Models, Matching Markets**

I. Introduction

The current cloud computing market structure is like an oligopoly with only a few big cloud providers. Due to economies of scale, these cloud providers can provide cost effective services by optimizing input resources such as power and real estate. However, such a service supply structure has inherently many drawbacks. For example, they put undue stress on power and other natural resources, disturbing the local micro-climates and eco-systems. Further, any failure of such centralized systems will have a widespread impact on innumerably many end users.

Instead, we can envisage a market structure with large number of distributed mini/micro cloud providers leading to a perfect competition between them. Further, the drawbacks of the existing system such as the impact on the local environments and few points of failure are mitigated substantially. However, there will be many consumer resource requests which individual cloud providers will not be able to cater due to shortage of resources. So multiple cloud providers have to come together and form a temporary coalition or a federation to serve such requests [1]–[4]. However, for seemless flow of services from cloud providers to end users and other value added services, intermediate brokers are necessary.

Motivated by this, in this paper, we propose a Multi-Cloud model market structure consisting of cloud providers, end users and intermediate brokers through which trade happens. The goods under consideration are virtual machines from the Infrastructure-as-a-Service (IaaS) layer. We formulate the problem of socially optimal allocation of virtual machines from cloud providers to end users via brokers as an integer linear programming problem. Then we show that a socially optimal allocation may have unstable producer-consumer[1] pairs who have an incentive to deviate from the current allocation. Then we propose an iterative distributed allocation algorithm with no centralized authority which eliminates such unstable producer-consumer pairs. Our algorithm can be viewed as a particular way of generalizing the stable matching algorithm due to Gale and Shapley [5].

The rest of the paper outline is as follows. Section II contains the Multi-Cloud marketplace structure and the relevant brokerage model; Section III defines social welfare and formulates the problem of optimal social welfare generation as an integer linear programming problem and proposes a greedy algorithm to solve it; Section IV contains the generalized distributed stable allocation algorithm and compares the social welfare generated by such an allocation with a centralized optimal allocation; Section V contains experimental results; Section VI contains the related work and finally we conclude in Section VII.

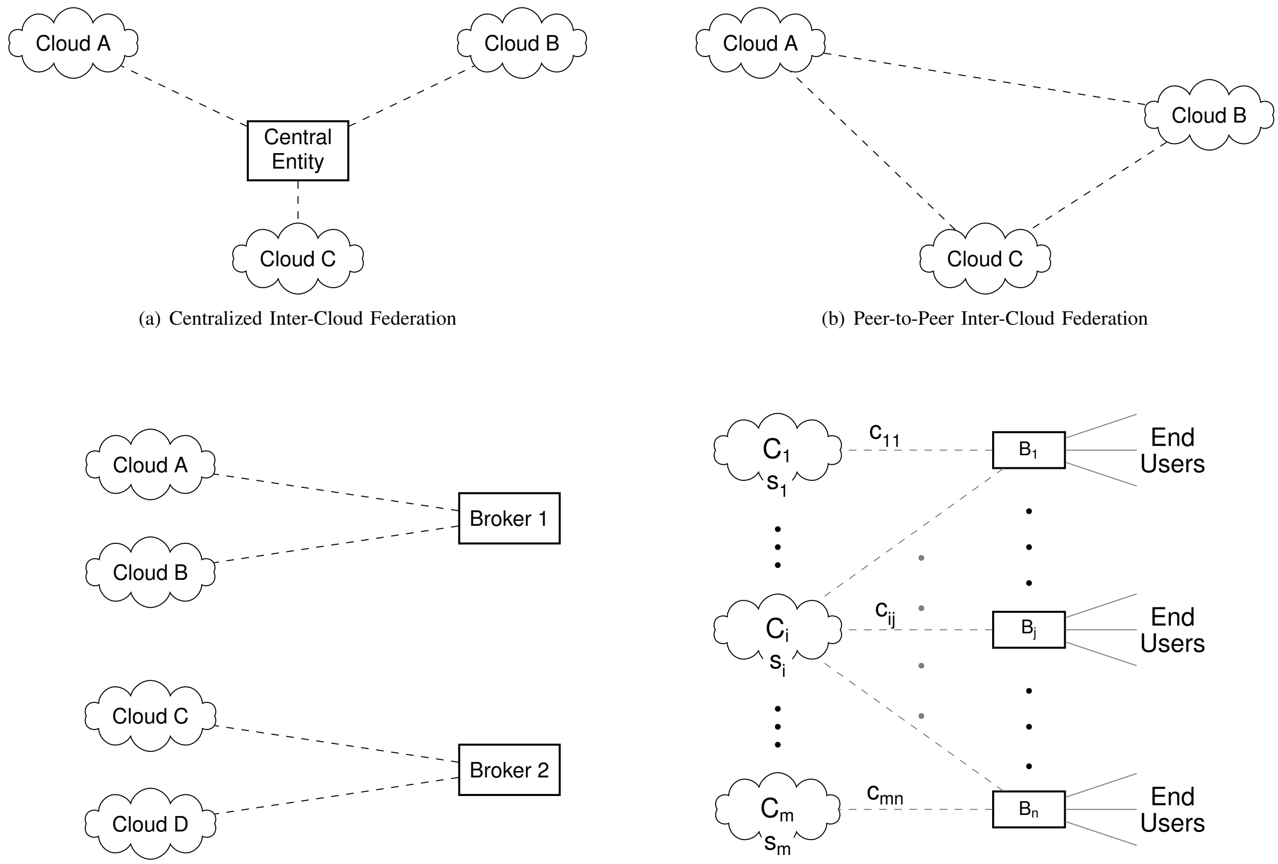
II. Multi-Cloud Marketplace Structure

In this section, we introduce the existing taxonomy of Inter-Cloud architectures and brokering mechanisms [4]; and show how the proposed Multi-Cloud marketplace structure extends that taxonomy.

An Inter-Cloud is a collection of cloud providers who can be either collaborating or competing with each other. When they

[1]Here cloud providers are producers and brokers are consumers. Although the virtual machines are ultimately consumed by end users, for the purpose of technical discussion, brokers are considered as consumers.

978-1-5386-5504-7/18/$31.00 ©2018 IEEE
DOI 10.1109/UCC.2018.00034



(a) Centralized Inter-Cloud Federation

(b) Peer-to-Peer Inter-Cloud Federation

(c) Muli-Cloud Model

(d) Extended Multi-Cloud Model

Fig. 1. Inter-Cloud and Multi-Cloud Models

work in a collaborative fashion, we call them as a *federation*. This collaboration can happen using a centralized entity or in a peer-to-peer fashion as depicted in Figures 1(a) and 1(b) respectively. When the cloud providers compete with each other, an end user typically needs a brokerage service to aggregrate resources in a cost optimal way, resulting in a Multi-Cloud model from Figure 1(c). In this model, the set of cloud providers served by a broker is disjoint from that of the other. For example, in Figure 1(c), no cloud provider gets simultaneously served by brokers B_1 and B_2. In this paper, we extend this model, by relaxing this constraint, and permit a cloud provider to perform trade with more than one broker, and vice-versa, as shown in Figure 1(d). In the rest of this section, we describe such a market place structure where virtual machines at the IaaS layer are traded and in subsequent sections we study trade optimality and stability.

We consider a marketplace consisting of m cloud providers $\mathcal{C} = \{C_1, \cdots, C_m\}$ and n brokers $\mathcal{B} = \{B_1, \cdots, B_n\}$. The cloud providers could be offering various classes of virtual machines such as *small*, *medium* and *large*. We restrict our attention to the trade of only a specific class of virtual machines for the sake of simplicity although our model can accommodate multiple virtual machine classes simultaneously. Each broker B_j quotes a rental price p_j for an instance of a virtual machine to an end user. We assume that an end user is hooked to a broker due to constraints such as trade restrictions arising from geo-political reasons, prior long term agreements, quality-of-service requirements, etc. Due to that, an end user cannot switch from a broker B_j to a broker B_k even if the price $p_k < p_j$. Let d_j be the number of virtual machine instances which the broker B_j would like to rent. Broker B_j would have arrived at the demand d_j by aggregating the requests of one or more end users subscribed to him. A cloud provider C_i offers a virtual machine instance to a broker B_j at price c_{ij}. Note that a cloud provider can offer a virtual machine instance of same type to different brokers at different prices, i.e., c_{ij} need not be equal to $c_{ij'}$ for $j \neq j'$. The cost differences could be due to geo-political economic reasons or rate contracts between cloud providers and brokers. Let s_i be the capacity or the supply constraint of the cloud provider C_i. Let $A_{m \times n}$ be an integer allocation matrix, such that the entry a_{ij} denote the number of virtual machines supplied by the cloud provider (producer) C_i to the broker (consumer) B_j. Let α_i be the cost incurred by the cloud provider C_i to manufacture a virtual machine. Figure 1(d) depicts the marketplace structure and Table I summarizes all the parameters of the model. In the following section, under the proposed brokerage model, we define the notions of producer and consumer surplus; and formulate the problem of maximizing the social welfare as an integer linear programming problem. Then from the objective function, derive a simple greedy algorithm to solve the maximization problem.

TABLE I
SUMMARY OF MODEL PARAMETERS.

Parameters	Description
$\mathcal{C} = \{C_1, \cdots, C_m\}$	Cloud providers.
$\mathcal{B} = \{B_1, \cdots, B_n\}$	Brokers
$C_{m\times n}$	Price matrix. Entry c_{ij} is the price at which C_i offers a virtual machine to B_j.
$\vec{p} = (p_1, \cdots, p_n)$	Broker B_j offers a virtual machine to an end user at a price p_j.
$\vec{d} = (d_1, \cdots, d_n)$	d_j denotes the aggregated virtual machine demand from broker B_j.
$\vec{s} = (s_1, \cdots, s_m)$	s_i denotes the available supply of virtual machines from cloud provider C_i.
$A_{m\times n}$	Allocation matrix. Entry a_{ij} is the quantity of virtual machines C_i allocates to B_j.
$\vec{\alpha} = (\alpha_1, \cdots, \alpha_m)$	α_i denotes the cost incurred by C_i to manufacture a virtual machine.

III. MAXIMIZING SOCIAL WELFARE

In this section we seek to find allocations that maximize the social welfare. Towards this end, for a particular allocation matrix A, we have the following definitions for the producer and consumer surpluses or profits. Recall that we consider the cloud providers as producers and brokers (not end users) as consumers.

Definition III.1. *The surplus CS_i of a cloud provider C_i is defined as*

$$CS_i = \sum_{j=1}^{n} (c_{ij} - \alpha_i) a_{ij}.$$

Definition III.2. *The surplus BS_j of a broker B_j is defined as*

$$BS_j = \sum_{i=1}^{m} (p_j - c_{ij}) a_{ij}.$$

Definition III.3. *The social welfare W of a marketplace is defined as the sum of all the producer and consumer surpluses, and is given by*

$$W = \sum_{i=1}^{m} CS_i + \sum_{j=1}^{n} BS_j. \tag{1}$$

The problem of maximizing the social welfare is expressed as an integer linear program, denoted by `opt`, as follows.

$$\texttt{opt}: \quad \max_{\{a_{ij}\}} W$$

$$\sum_{j=1}^{n} a_{ij} \leq s_i, \; i = 1, \cdots, m \tag{2}$$

$$\sum_{i=1}^{m} a_{ij} \leq d_j, \; j = 1, \cdots, n \tag{3}$$

$$a_{ij} \geq 0, \; a_{ij} \text{ is an integer} \quad i = 1, \cdots, m, \quad j = 1, \cdots, n \tag{4}$$

Constraints (2) and (3) correspond to the restrictions on the supply and demand respectively. We can do an algebraic simplification of the social welfare function as follows.

$$W = \sum_{i=1}^{m} CS_i + \sum_{j=1}^{n} BS_j \tag{5}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n} (c_{ij} - \alpha_i) a_{ij} + \sum_{j=1}^{n}\sum_{i=1}^{m} (p_j - c_{ij}) a_{ij} \tag{6}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n} (p_j - \alpha_i) a_{ij} \tag{7}$$

The above integer linear program `opt` is equivalent to a maximization version of an unbalanced transportation problem. There exists polynomial time algorithms which give optimal integral solutions for the transportation problems [6]. However, the simplified objective function (7) directly inspires a greedy algorithm without invoking the transportation theory. The main idea is that, the social welfare can be maximized by transporting goods (virtual machines) from producer sources with low production costs (α_i) to consumer destinations which offer high prices (p_j). Hence, we can use a greedy strategy wherein in each iteration of the algorithm virtual machines are traded between a producer-consumer pair (C_i, B_j). The producer-consumer pairs (C_i, B_j) are considered in the descending order of their $p_j - \alpha_i$ values. A producer-consumer pair performs maximal possible trade in an iteration which is determined by their residual supply and demand. If s_i^* and d_j^* denote the residual supply and demand, then $a_{ij} = min(s_i^*, d_j^*)$ and $s_i^* = s_i^* - a_{ij}$ and $d_j^* = d_j^* - a_{ij}$. Since, we are working on an maximization problem and all the problem parameters, $\{a_{ij}\}$, are positive integers, it is not possible to simultaneously have a cloud provider C_i with residual supply and a broker B_j with residual demand unless $p_j < \alpha_i$.

If we assume that the cost of producing a virtual machine is the same, α, for all the cloud providers, then Equation (7) assumes the following form.

$$W = \sum_{j=1}^{n}\sum_{i=1}^{m} (p_j - \alpha) a_{ij} \tag{8}$$

$$= \sum_{j=1}^{n} (p_j - \alpha) \sum_{i=1}^{m} a_{ij} \tag{9}$$

If there is a feasible way to meet all the broker demands, then there exists an allocation such that $\sum_{i=1}^{m} a_{ij} = d_j$ for all $1 \leq j \leq n$. Then the optimal value of the welfare function is given as follows.

$$\boxed{W = \sum_{j=1}^{n} (p_j - \alpha) d_j} \tag{10}$$

There can be many allocations meeting complete broker demands and hence generate the optimal social welfare value, but each allocation will yield different profits to the individual cloud providers and brokers. In the next section, we define the notion of a stable allocation and propose a polynomial time

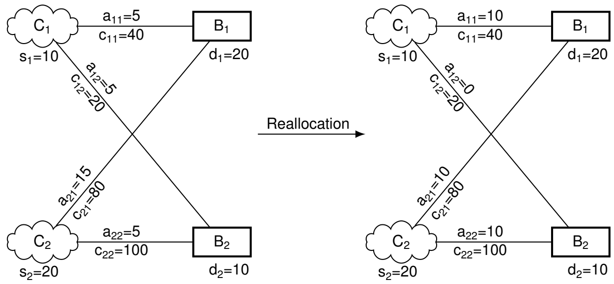


Fig. 2. The allocation on the left is unstable and hence shifts to a stable allocation depicted on the right.

distributed algorithm for the same by generalizing the Gale-Shapley stable matching algorithm [5].

Multiple Virtual Machine Types If there are multiple virtual machine types like *small*, *medium* and *large* available for trade, then the social welfare optimization problem and the corresponding greedy algorithm can be easily extended, as the market for each virtual machine type is independent of the other in our model. We do not present that in this paper as it introduces more notation with no additional insight on the problem.

IV. Distributed Stable Allocation using Generalized Gale-Shapley

An allocation A is *feasible* if it satisfies the supply constraint (2) and the demand constraint (3). A feasible allocation is *optimal* if it maximizes the social welfare function (1). Inspired by the stable marriages and college admissions problems due to Gale-Shapley [5], we define the notion of a stable allocation as follows.

Definition IV.1. *We say that a feasible allocation A is unstable if it consists of cloud providers $C_{i'}$ and C_i, and brokers B_j and $B_{j'}$ satisfying the following properties.*

1) $a_{ij'} > 0$ *and* $a_{i'j} > 0$.
2) C_i *prefers* B_j *to* $B_{j'}$ *since* $c_{ij} > c_{ij'}$.
3) B_j *prefers* C_i *to* $C_{i'}$ *since* $c_{ij} < c_{i'j}$.

Note that the price (p_j) at which a broker offers virtual machines to end users do not figure in the definition of the stability. This is due to the fact that the selling price is fixed independent of the cost (c_{ij}) at which a broker procures the virtual machines. So, to maximize his profit a broker prefers cloud providers who offer the best prices.

In an unstable allocation, as per Definition IV.1, the cloud provider C_i and broker B_j have an incentive to deviate from the current current allocation structure in a mutually beneficial manner by respectively improving their individual surplus. For example, consider a model with cloud providers $\{C_1, C_2\}$ and two brokers $\{B_1, B_2\}$. Let the supply and demand vectors be $\vec{s} = (10, 20)$ and $\vec{d} = (20, 10)$ respectively. Let the price matrix be $C = \left(\begin{smallmatrix} 40 & 20 \\ 80 & 100 \end{smallmatrix}\right)$ and the current feasible allocation matrix be $A = \left(\begin{smallmatrix} 5 & 5 \\ 15 & 5 \end{smallmatrix}\right)$. Then the surplus of cloud providers C_1 and C_2 is given as follows:

$$
\begin{aligned}
CS_1 &= 5(40 - \alpha_1) + 5(20 - \alpha_1) \\
&= 300 - 10\alpha_1 \\
CS_2 &= 15(80 - \alpha_2) + 5(100 - \alpha_2) \\
&= 1700 - 20\alpha_2
\end{aligned}
$$

Similarly the surplus of the brokers B_1 and B_2 is given as follows:

$$
\begin{aligned}
BS_1 &= 5(p_1 - 40) + 15(p_1 - 80) \\
&= 20p_1 - 1400 \\
BS_2 &= 5(p_2 - 20) + 5(p_2 - 100) \\
&= 10p_2 - 600
\end{aligned}
$$

We can notice that the cloud provider C_1 would be happy to move the current supply of 5 virtual machines from broker B_2 to B_1 as it increases her profit, and the broker B_1 would also be happy to accept the same offer. Then broker B_2 is forced to procure the residual demand of 5 virtual machines from cloud provider C_2. This leads to the new allocation matrix $A' = \left(\begin{smallmatrix} 10 & 0 \\ 10 & 10 \end{smallmatrix}\right)$. Figure 2 summarizes this example. Note that this process of reallocation benefits cloud provider C_2 also but upsets the broker B_2 as his surplus reduces to $10(p_2 - 100)$.

One-to-One Stable Marriages Problem In the stable marriages problem, there are equal number of men and women, with each man having a preference list of women, and vice-versa. The problem is to find a matching between men and women such that there exists no man and woman who prefer each other to their current partners. Such a matching is called a stable matching. We can model this problem under our framework by considering men as cloud providers with a single unit of supply capacity and women as brokers with single unit of demand. The price matrix C naturally induces a preference list for everyone.

One-to-Many College Admissions Problem In the college admissions problem, we have to construct a one-to-many match between colleges and students such that there does not exist a college and a student who prefer each other than to their current assignment. This problem can viewed in our framework by considering the colleges as cloud providers with supply capacity equal to their admission strength and students as brokers with unit demand.

Many-to-Many Stable Matching Our problem formulation is a generalization of the aforementioned market scenarios summarized as follows.

1) A collection of producers (cloud providers) with respective supply capacities.
2) A collection of consumers (brokers) with respective demand.
3) A variable price structure inducing a natural preference list for every producer and consumer.

We can see that the proposed model captures many market scenarios and the particular multi-cloud marketplace is a specific instantiation of the model. Further, if there are multiple independent goods for trade, then we can model problem by considering the marketplace for each good as an independent entity. However, if there is interdependence between goods in terms of price variation due to bundling etc., our current model is not powerful enough to handle.

A. Stable Allocation Algorithm

In this section, we propose a distributed algorithm for stable allocation of virtual machines from the cloud providers to the brokers.

The algorithm is iterative in nature with each iteration consisting of a `BPhase` (broker phase) followed by a `CPhase` (cloud provider phase).

1) `BPhase`: Each broker chooses a previously unapproached cloud provider who maximizes his surplus. Then he bids for resources from him based on his residual demand.
2) `CPhase`: Each cloud provider evaluates the proposed bids and responds to them in such a way that maximizes his surplus. If a cloud provider gets a better bid in a later iteration, he may revoke already allocated resources partially or fully from one or more brokers.

The algorithm terminates when there are no more bids from the brokers, either because their demands are met or they exhausted all the available cloud providers. A more detailed explanation of the algorithm follows.

Let $\overrightarrow{rs}$ and $\overrightarrow{rd}$ denote the residual supply and demand at the cloud providers and the brokers respectively. Initially $\overrightarrow{rs}$ and $\overrightarrow{rd}$ are equal to $\vec{s}$ and $\vec{d}$ respectively. A broker B_j would like to maximize his profit $\Sigma_{i=1}^{m}(p_j - c_{ij})a_{ij}$. Hence, he prepares a preference list of cloud service providers in the decreasing order of $p_j - c_{ij}$ values and excludes those cloud providers for which $p_j < c_{ij}$. Since p_j is constant, a broker arrives at the same preference by considering increasing c_{ij} values with ties broken arbitrarily. A broker attempts to meet his residual demand iteratively by approaching the cloud providers one at a time based on his preference list. Similarly, a cloud provider C_i tries to maximize his profit $\Sigma_{j=1}^{n}(c_{ij} - \alpha_i)a_{ij}$. So, he would like to first cater to the demands of a broker which maximizes $c_{ij} - \alpha_i$. Since the production costs of a cloud service provider C_i are assumed to be constant, he prefers a broker B_j over $B_{j'}$ if $c_{ij} > c_{ij'}$. Based on this observation, every cloud provider arrives at a preference list of brokers by breaking ties arbitrarily. We now describe the `BPhase` and `CPhase` of each iteration of the algorithm.

`BPhase`: Each broker B_j makes a request for the supply of rd_j (residual demand) virtual machines to the cloud provider who is at the head of his current preference list. A broker makes a request to a cloud provider only once. Hence, after the request, the cloud service provider at the head of the preference list is discarded.

`CPhase`: Each cloud provider C_i accumulates all the requests from various brokers. Let B_p^i be the set of all brokers who approached C_i during the past and the current round of allocation. Let $\vec{a_i}$ denote the i^{th} row of the allocation matrix A. Then C_i readjusts the current allocation $\vec{a_i}$ in such a way that the brokers get as much supply as they requested based on the preference order upto the extent permitted by the capacity constraint. During this process, supply committed to a previous broker B_j could be revoked fully or partially and re-allocated to a new broker $B_{j'}$ if $B_{j'}$ is preferred over B_j. If supply s is revoked from B_j, then it will be communicated to B_j and B_j will adjust his residual demand accordingly.

The algorithm terminates when no broker makes a new supply request during a `BPhase`(). A broker will not make a supply request during an iteration if either there is no residual demand or he exhausted all the cloud providers in his preference list.

Theorem IV.1. *The proposed allocation algorithm converges in $O(mn)$ iterations in the worst case.*

Proof: During the `BPhase` of an iteration, if the residual demand of all the brokers is zero, then the algorithm terminates. Otherwise, there exists a broker whose residual demand is non-zero and he bids for virtual machines from the cloud provider at the head of his preference list. Since a broker considers a cloud service provider only once and the maximum size of sum of the preference lists from all the brokers is mn, the algorithm takes $O(mn)$ iterations to terminate in the worst case. ■

Theorem IV.2. *There exists no unstable pair after the termination of the allocation algorithm.*

Proof: Let there exists an unstable pair (C_i, B_j). Then from the Definition IV.1, we can infer that there exists a cloud provider $C_{i'}$ and a broker $B_{j'}$ such that:

1) $a_{ij'} > 0$.
2) C_i prefers B_j to B'_j.
3) $a_{i'j} > 0$.
4) B_j prefers C_i to $C_{i'}$.

That means C_i is willing to move some supply s from $B_{j'}$ to B_j to improve his surplus. Similarly, B_j is willing to get that supply from C_i instead of $C_{i'}$. However, B_j would have already tried to get that much supply or even more from C_i during the allocation algorithm since C_i occurs in his preference list ahead of $C_{i'}$. Similarly, C_i would have revoked the promised supply to $B_{j'}$ if necessary and reallocated the same to B_j during the allocation algorithm. Hence such unstable pairs does not exist after the termination of the allocation algorithm. ■

B. Discussion

Consider the stable marriages problem, where there are n men an n women. Each man has a preference list of women and each woman has a preference list of men. We call a matching between men and women unstable if there exists a man and woman pair who prefer each other to their existing partners. If no such unstable pairs exist, then the matching is stable.

For a man m (woman w), let p_m^* (p_w^*) denote the best ranked partner among all the possible stable matchings. In the Gale-Shapley algorithm, where men are the proposers, the resulting

TABLE II
FIRST ROW GIVES THE BASE HOURLY RENTAL PRICE OF `M4.10XLARGE` INSTANCES. THE REST OF THE TABLE IS THE TRANSPOSE OF THE COST MATRIX C.

Location	USA N. Virginia (C_1)	USA Ohio (C_2)	USA N. California (C_3)	USA Oregon (C_4)	USA AWS GovCloud (C_5)	EU Frankfurt (C_6)	EU Ireland (C_7)	Asia Singapore (C_8)	Asia Tokyo (C_9)	Asia Sydney (C_{10})	Asia Seoul (C_{11})	Asia Mumbai (C_{12})	SA Sao Paulo (C_{13})
Base Price	\$2.39	\$2.39	\$2.79	\$2.39	\$3.02	\$2.85	\$2.64	\$3.55	\$3.47	\$3.36	\$3.30	\$3.38	\$3.43
US (B_1) (p_1=\$4.02)	\$2.39	\$2.39	\$2.79	\$2.39	\$3.02	\$3.71	\$3.43	\$4.80	\$4.70	\$4.54	\$4.46	\$4.56	\$4.39
EU (B_2) (p_2=\$4.14)	\$3.11	\$3.11	\$3.63	\$3.11	\$3.93	\$2.85	\$2.64	\$4.58	\$4.49	\$4.34	\$4.26	\$4.35	\$4.52
Asia (B_3) (p_3=\$3.91)	\$3.23	\$3.23	\$3.77	\$3.23	\$4.08	\$3.68	\$3.41	\$3.55	\$3.47	\$3.36	\$3.30	\$3.38	\$4.49
SA (B_4) (p_4=\$4.20)	\$3.06	\$3.06	\$3.58	\$3.06	\$3.87	\$3.76	\$3.49	\$4.65	\$4.55	\$4.41	\$4.33	\$4.42	\$3.43

stable matching is called as *men-optimal*, since every man m gets the best possible ranked woman partner p_m^*. Similar thing happens when woman are the proposers resulting in *women-optimal* stable matching.

In the current work, we can arrive at analogous definitions and analyze whether the proposed algorithm is *consumer-optimal* (broker optimal) or *producer-optimal* (cloud-provider optimal).

Definition IV.2. *Let CS_i^* denote the maximum surplus of the cloud provider C_i among all the stable allocations possible.*

Definition IV.3. *Let BS_j^* denote the maximum surplus of the broker B_j among all the stable allocations possible.*

In the distributed stable allocation algorithm proposed in the previous section, brokers are the proposers. We can easily envisage an algorithm where cloud providers are the proposers. We make the following conjecture in this paper.

Conjecture: *In the stable allocation algorithm, when brokers are the proposers, each broker B_i gets the maximum possible surplus BS_j^* from among all the stable allocations.*

An equivalent conjecture can be made when cloud providers are the proposers. Even if the conjecture in this strong form is not true, it is possible that when brokers are the proposers, they may get an advantage when compared with cloud providers. We present an empirical analysis of this claim in the experimental results section.

C. Impact on Social Welfare

In this section, we will show that when we use stable matching algorithm for allocation, the generated social welfare can be suboptimal when compared with a centralized allocation strategy obtained by solving the optimization problem `opt` from Section III. The ratio between these values, which indicates the social cost of selfishness, is conventionally called as *price of anarchy* [7].

When there is sufficient supply from cloud providers to meet the demands of the brokers, then from Equation (10) we can infer that the social welfare is independent of resource allocation. So both centralized and, distributed allocation using stable matching, results in the same social welfare value. However, when demand is more than supply, then the social welfare values may differ as the following example illustrates.

imp

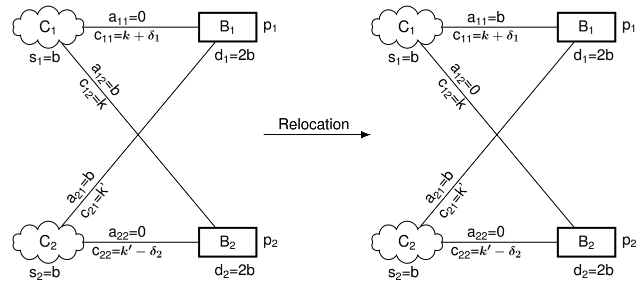


Fig. 3. The unstable allocation on the left generates more social welfare than the stable allocation on the right.

Figure 3 illustrates a scenario with two cloud providers and two brokers with supply and demand vectors (b, b) and $(2b, 2b)$ respectively. Let the price matrix be $C = \begin{pmatrix} k+\delta_1 & k \\ k' & k'-\delta_2 \end{pmatrix}$ for some constants $k, k', \delta_1, \delta_2 > 0$ and $k' > \delta_2$. The allocation on the left side of Figure 3 is given by $A = \begin{pmatrix} 0 & b \\ b & 0 \end{pmatrix}$. The social welfare that is generated due to this allocation is

$$\begin{aligned} W &= CS_1 + CS_2 + BS_1 + BS_2 \\ &= b(k - \alpha_1) + b(k' - \alpha_2) + b(p_1 - k') + b(p_2 - k) \\ &= b((p_1 + p_2) - (\alpha_1 + \alpha_2)). \end{aligned}$$

We can notice that in the current allocation cloud provider C_1 prefers broker B_1 over B_2 since $c_{11} > c_{12}$ by δ_1 margin. Also, B_1 has still a residual demand of b. This makes the current allocation unstable and as a result supply reallocation happens wherein cloud provider C_1 moves b units of virtual machines to broker B_1 from broker B_2 assuming $p_1 > k + \delta_1$. Then the new allocation matrix would be $A' = \begin{pmatrix} b & 0 \\ b & 0 \end{pmatrix}$. This affects the surpluses of cloud providers and brokers in the following way. Figures 3 depicts the unstable and stable allocations.

$$\begin{aligned} CS_1^{old} &= b(k - \alpha_1) &\rightarrow\quad CS_1^{new} &= b(k + \delta_1 - \alpha_1) \\ CS_2^{old} &= b(k' - \alpha_2) &\rightarrow\quad CS_2^{new} &= b(k' - \alpha_2) \\ BS_1^{old} &= b(p_1 - k') &\rightarrow\quad BS_1^{new} &= b(p_1 - (k + \delta_1)) \\ & & &\quad + b(p_1 - k') \\ BS_2^{old} &= b(p_2 - k) &\rightarrow\quad BS_2^{new} &= 0 \end{aligned}$$

The social welfare generated using the new stable allocation matrix A' is

$$\begin{aligned} W_{stable} &= CS_1 + CS_2 + BS_1 + BS_2 \\ &= b(k + \delta_1 - \alpha_1) + b(k' - \alpha_2) \\ &\quad + b(p_1 - (k + \delta_1)) + b(p_1 - k') \\ &= b(2p_1 - (\alpha_1 + \alpha_2)) \end{aligned}$$

If we consider the ratio of the social welfare generated in the two cases, we have

$$\frac{W}{W_{stable}} = \frac{b((p_1 + p_2) - (\alpha_1 + \alpha_2))}{b(2p_1 - (\alpha_1 + \alpha_2))}.$$

We can see from above, if $p_2 \to \infty$, then $W \to \infty$ but W_{stable} remains constant. This is the social cost of selfishness of the cloud provider C_2 and broker B_1.

V. Experimental Results

In this section, we analyze the surpluses of brokers and cloud providers, and the overall social welfare generated by using centralized allocation vis-as-vis stable matching based allocation. For the experimental analysis, we consider a market with 13 cloud providers and 4 brokers. Each of the 13 different Amazon EC2 regions are considered as distinct cloud providers. The geographical location of the cloud providers is as follows: C_1 to C_5 from USA, C_6 and C_7 from European Union (EU), C_8 to C_{12} from Asia, and C_{13} from South America (SA). The four brokers B_{usa}, B_{eu}, B_{asia} and B_{sa} are from the regions USA, EU, Asia and SA respectively. We consider the trade of only one kind of a virtual machine `m4.10xlarge`. As discussed at the end of Section III, our model can handle the trade of multiple virtual machine types but here we consider only one virtual machine type for the sake of simplicity. Each cloud provider has a base price for a single instance of `m4.10xlarge` type virtual machine (refer Table II). A cloud provider offers a virtual machine to a broker within the same geographical region at its base price. But if the broker is from a different region he charges a higher rate. The price hike percentage based on geographical location is provided in the Table III. For example, a cloud provider from USA charges 30 percent extra on its base price to the broker B_{eu}. A broker B_j offers a virtual machine to an end user at a price

$$\boxed{p_j = 1.1 \times \text{avg}\{c_{ij}\}}$$

It is considered as such to assume a realistic scenario that a broker is unable to purchase resources from some of the cloud providers due to economic constraints. Thus, it is possible that although there is a residual demand as well as residual supply, yet no trade happens. For ease of interpreting the experimental results, we set the production costs of virtual machines to zero for all the cloud providers.

A. Comparison of Social Welfare

We fix the supply of each of the 13 cloud providers to ten virtual machines. We generate 200 demand vectors such that the demand-supply ratio falls in the range $(0, 3)$. Figure 4(a) plots the corresponding social welfare values for various demand instances. It can be noted that the social welfare due to stable allocation is always upper bounded by that of the optimal allocation as expected. The difference in the social welfare value between optimal and stable allocations depends on factors such as cost matrix C and price vector $\vec{p}$ (refer Table I and Equation (1)). In order to get a better understanding on how stable allocation compares with optimal allocation in a manner oblivious to these factors, we define the following utility functions and the associated social welfare functions based on these utilities. Recall that there m cloud providers and n brokers in our setting.

Definition V.1. *Consider a cloud provider C_i and a broker B_j. Let k be the rank of the broker in the preference list of the cloud provider and l be the rank of the cloud provider in the preference list of the broker. When the cloud provider and the broker trades a virtual machine, then their respective utilities are defined as $U_c(i) = n - k + 1$ and $U_b(j) = m - l + 1$.*

Definition V.2. *The overall utility associated with an allocation matrix A is defined as follows*

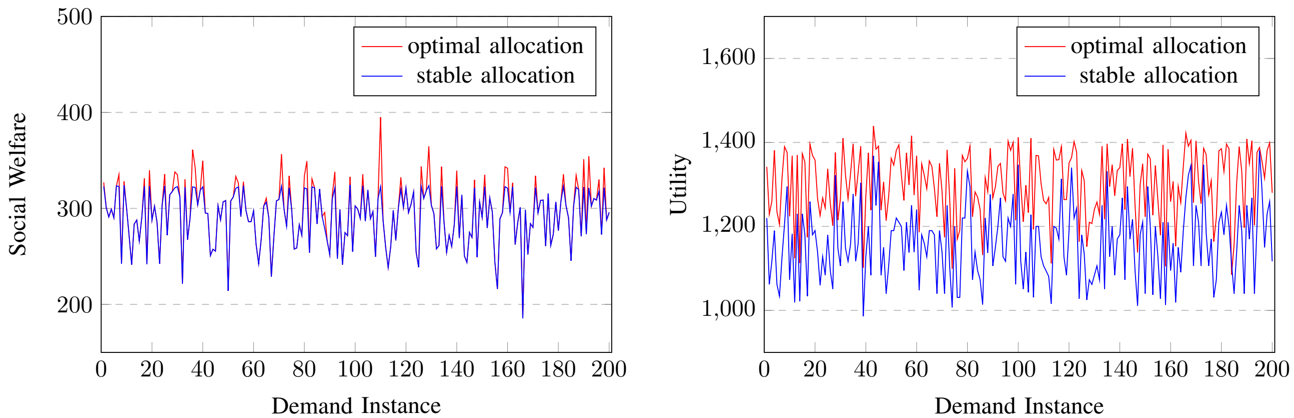
$$U(A) = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}U_c(j)\right) + \sum_{j=1}^{n}\left(\sum_{i=1}^{m} a_{ij}U_b(i)\right).$$

Figure 4(b) compares the overall utility generated using optimal and stable allocations. It can be noticed that the average utility due to optimal and stable allocations is around 1300 and 1100. Further, the utility due to optimal allocation is always greater than that of due to stable allocation as expected.

Figure 5 shows the social welfare generated for a given demand-supply ratio. It can be noted that when the demand-supply ratio is small (<0.75), the social welfare generated due to the stable allocation almost matches that of the optimal allocation. As the demand is less, there is not a stiff competition between the brokers for procurement of resources. However, as the demand increases, the competition between brokers increase for the available resources. The amount of trade is decreased because of stability and economic contraints. However, average welfare ratio increases as the demand increases significantly. It is attributed to the fact that there is an individual increase in demand of each broker. Since each broker has a high demand, the trade increases with respect to a broker who can afford more resources, thereby increasing total trade.

TABLE III
Percentage price hikes adopted by cloud providers based on geographical regions.

Cloud Providers	$B_1(USA)$	$B_2(EU)$	$B_3(Asia)$	$B_4(SA)$
C_1-C_5 (USA)	0	30	35	28
C_6, C_7 (EU)	30	0	29	32
C_8-C_{12} (Asia)	35	29	0	31
C_{13} (SA)	28	32	31	0



(a) Social welfare due to optimal and stable allocations.

(b) Overall utility generated using optimal and stable allocations.

Fig. 4. Comparison of social welfare and overall utility generated due to optimal and stable allocations.

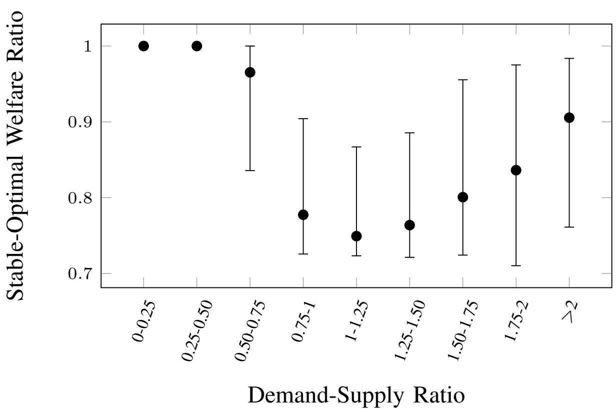


Fig. 5. Stable-optimal social welfare ratio as against demand-supply ratio.

B. Comparison of Broker Surplus and Cloud Provider Surplus

We compare the surpluses of brokers and cloud providers when the demand is just greater than supply as that is the case when the social welfare generated from the stable allocation is substantially less than that of the optimal allocation, from Figure 5. We generate 200 instances of the demand vector from a normal distribution with mean 40 and standard deviation 10. The average surpluses of cloud providers and brokers for optimal and stable allocations are given in Figures 6(a) and 6(b) respectively.

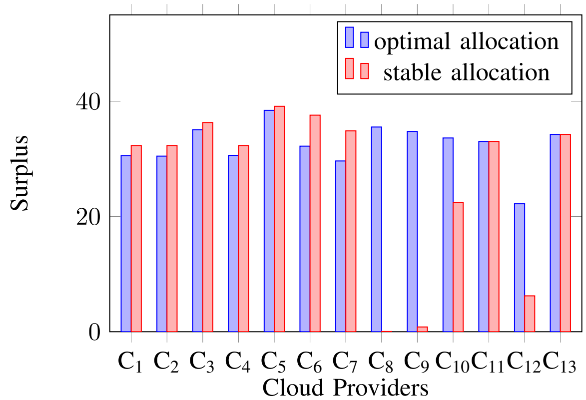
It can noticed that cloud providers C_8 and C_9 hardly get any surplus in the stable allocation as they are less preferred by all the brokers due to their highest base price (refer Table II). Cloud provider C_{13} makes good surplus inspite of the high base price as he comes fourth in the preference list of the local broker B_4 and the cloud providers occuring earlier in B_4 list prefer other high paying brokers to B_4. With respect to brokers, broker B_1 has almost no surplus in stable allocation as every bid he makes to a cloud provider(which he can afford), the corresponding cloud provider gets a better bid from an alternative broker. Similar is the case for broker B_2 in most of the iterations of the algorithm.

C. Broker versus Cloud Provider Optimal Allocations

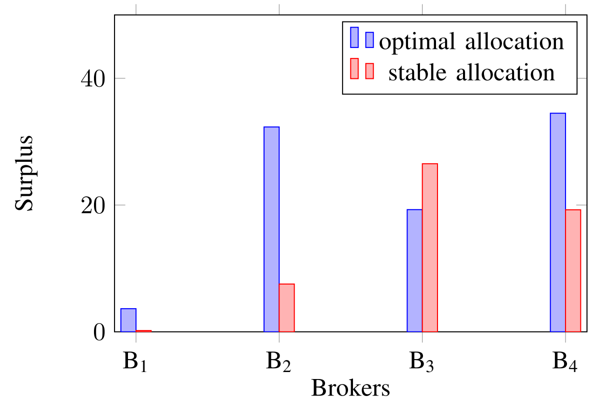
In this Section, we present the discussion from the viewpoint of a broker. An analgous discussion holds with respect to a cloud provider too. In Section IV, we made a conjecture that if the brokers are proposers in the allocation algorithm, then each broker gets the maximum possible surplus in any stable allocation. In order validate this conjecture empirically, we consider a setting in which 13 cloud providers and 4 brokers are present. The preference list for each cloud provider and broker is randomly chosen. Then we considered the following three cases: demand equals supply, demand is greater than supply and demand is less than supply. In all these three cases, we computed the utility of cloud providers and brokers. In each case, we compare the utilities when the proposers are brokers and cloud providers. The comparative utilities are shown in the Figures 7, 8 and 9. It can be noticed that in all the three cases, if brokers propose then they achieve greater utility when compared with the allocation where cloud providers propose. However, we do note that the experimental results does not prove that the achieved utility is the maximum possible from among all the stable allocations.

VI. Related Work

Assis and Bittencourt [2], Grozev and Buyya [4], Toosi et al. [1] provide an excellent survey on various cloud federation models and architectures. The marketplace model we proposed is based on the *aggregated broker service model* [1] and Inter-Cloud federation framework [2], [8]. Mashayeky et al. [9] and Khandelwal et al. [10] considered an Inter-Cloud federation model wherein the individual cloud providers cooperate with each other through a broker to meet end user requests for virtual machines of various types given their resource constraints. The profit generated by servicing a consumer request is distributed proportionately among the participating cloud providers using Banzhaf value measure. Broker is purely

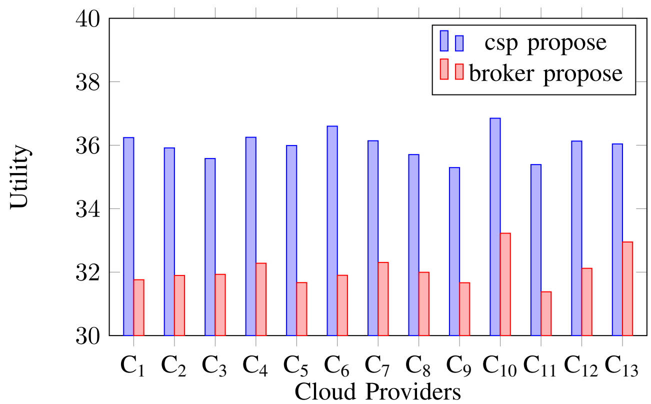


(a) Cloud provider surplus

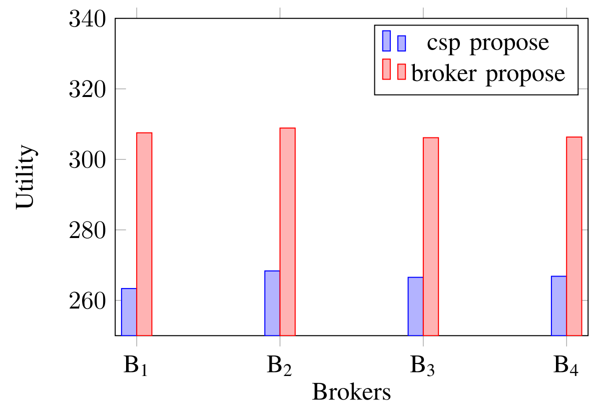


(b) Broker surplus

Fig. 6. Surpluses of cloud providers and brokers when demand to supply ratio is between 1 and 1.25

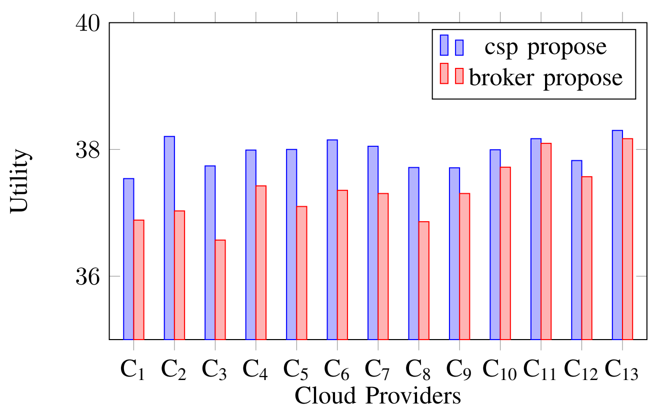


(a) Cloud provider Utility

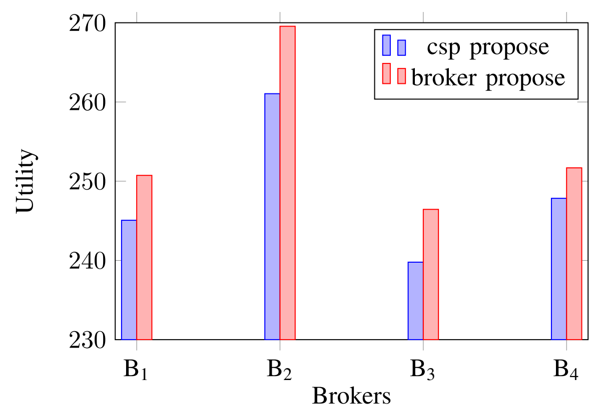


(b) Broker Utility

Fig. 7. Utility of cloud providers and brokers when demand equals supply



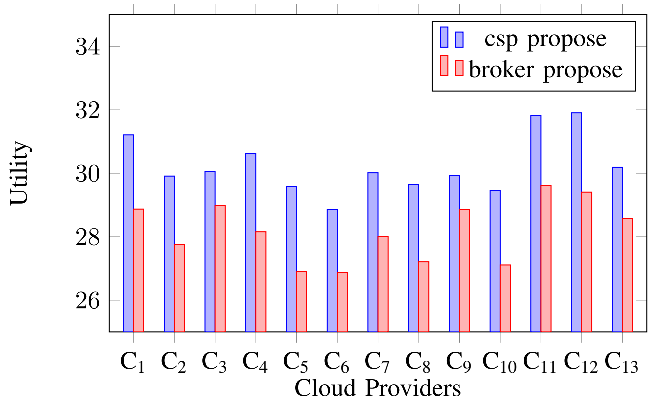
(a) Cloud provider Utility



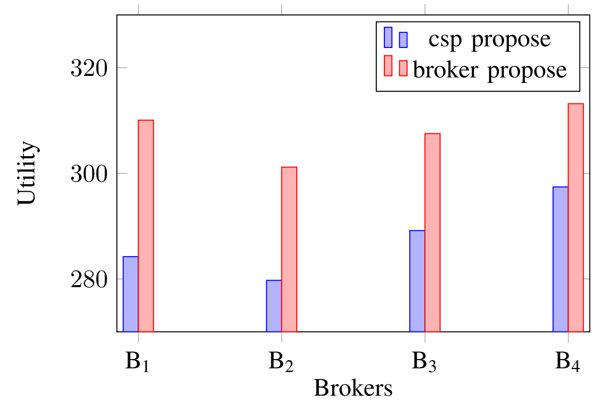
(b) Broker Utility

Fig. 8. Utility of cloud providers and brokers when demand is greater than supply

an altruistic intermediary in thier model. Whereas in our model, broker is a business entity who tries to maximize his profits. Hence, there is no profit sharing mechanism and cloud providers quote a price over and above their virtual machine production costs for their profits which will be earned directly. Our current work is at the IaaS layer but there is substantial work on the application layer brokering and enabling software architectures [11]–[15].



(a) Cloud provider Utility



(b) Broker Utility

Fig. 9. Utility of cloud providers and brokers when demand is less than supply

Caton et al. [16] proposed using stable matching algorithm to share resources between peers on a social compute cloud. The market structure on a social compute cloud is quite different from that of what we considered in this paper. In social compute clouds, trade typically happens in single unit (or very few unit) resources, whereas in the multi-cloud market structure of this paper, trade happens in large volume and hence the basic Gale-Shapley algorithm cannot be applied as it is. Further, our formulation clearly defines the social welfare function and how it is affected by centralized optimal allocation vis-a-vis stable allocation. Sim [17] proposed a broad framework identifying problems and challenges in market models involving cloud providers, brokers and end users. Our paper studies the interactions between cloud providers and brokers while discounting the effect of end users on the whole market structure.

VII. Conclusions

In this paper, we considered a cloud market place consisting of cloud providers, brokers and end consumers. The market is modeled using suitable supply and demand constraints; pricing models for cloud providers and brokers. Then we showed that a socially optimal resource allocation may not result in market equilibrium and proposed a generalization of Gale-Shapley stable matching algorithm which need not be socially optimal but produces stable allocations. Although, we proposed this generalization in the context of cloud computing, it is applicable in any producer-consumer market place.

References

[1] A. N. Toosi, R. N. Calheiros, and R. Buyya, "Interconnected cloud computing environments: Challenges, taxonomy, and survey," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 7:1–7:47, May 2014.

[2] M. Assis and L. Bittencourt, "A survey on cloud federation architectures: Identifying functional and non-functional properties," *Journal of Network and Computer Applications*, vol. 72, pp. 51 – 71, 2016.

[3] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *Int. J. High Perform. Comput. Appl.*, vol. 15, no. 3, pp. 200–222, Aug. 2001.

[4] N. Grozev and R. Buyya, "Inter-cloud architectures and application brokering: taxonomy and survey," *Software: Practice and Experience*, vol. 44, no. 3, pp. 369–390, 2014.

[5] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962. [Online]. Available: http://www.jstor.org/stable/2312726

[6] U. Brenner, "A faster polynomial algorithm for the unbalanced hitchcock transportation problem," *Oper. Res. Lett.*, vol. 36, no. 4, pp. 408–413, Jul. 2008.

[7] T. Roughgarden, *Selfish Routing and the Price of Anarchy*. The MIT Press, 2005.

[8] M. Makkes, C. Ngo, Y. Demchenko, R. Strijkers, R. Meijer, and C. de Laat, "Defining intercloud federation framework for multi-provider cloud services integration," in *The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization*, 2013, pp. 185–190.

[9] L. Mashayekhy, M. M. Nejad, and D. Grosu, "Cloud federations in the sky: Formation game and mechanism," *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 14–27, Jan 2015.

[10] Y. Khandelwal, S. Purini, and P. V. Reddy, "Fast algorithms for optimal coalition formation in federated clouds," in *Proceedings of the 9th International Conference on Utility and Cloud Computing*, ser. UCC '16, 2016, pp. 156–164.

[11] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling—l of application services," in *Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing - Volume Part I*, ser. ICA3PP'10, 2010, pp. 13–31.

[12] E. Carlini, M. Coppola, P. Dazzi, L. Ricci, and G. Righetti, "Cloud federations in contrail," in *Proceedings of the 2011 International Conference on Parallel Processing*, ser. Euro-Par'11, 2012, pp. 159–168.

[13] A. C. Marosi, G. Kecskeméti, A. Kertész, and P. Kacsuk, "Fcm: an architecture for integrating IaaS cloud systems," in *Proc. of the Second International Conference on Cloud Computing, GRIDs, and Virtualization (Cloud Computing 2011)*, 2011, pp. 7–12.

[14] D. Petcu, C. Crăciun, M. Neagul, S. Panica, B. Di Martino, S. Venticinque, M. Rak, and R. Aversa, "Architecturing a sky computing platform," in *Proceedings of the 2010 International Conference on Towards a Service-based Internet*, ser. ServiceWave'10, 2011, pp. 1–13.

[15] D. Bernstein, D. Vij, and S. Diamond, "An intercloud cloud computing economy - technology, governance, and market blueprints," in *Proceedings of the 2011 Annual SRII Global Conference*, ser. SRII '11, 2011, pp. 293–299.

[16] S. Caton, C. Haas, K. Chard, K. Bubendorfer, and O. F. Rana, "A social compute cloud: Allocating and sharing infrastructure resources via social networks," *IEEE Transactions on Services Computing*, vol. 7, no. 3, pp. 359–372, 2014.

[17] K. M. Sim, "Towards complex negotiation for cloud economy," in *International Conference on Grid and Pervasive Computing*. Springer, 2010, pp. 395–406.