Cost-efficient Resource Allocation for Decentralized Clouds

Hao Zhuang
LSIR, I&C, EPFL

Abstract—Cloud computing is gaining strength in the IT marketplace because of its on-demand resource provisioning, high availability and elasticity. To offer these advantages, many largely centralized computing infrastructures have emerged, which results in many concerns such as resource over-provisioning, huge energy consumption, privacy protection, data security, etc. In this proposal, we envision a decentralized cloud in which the resources are provided and shared by a group of small data centers. Through cooperations among small cloud vendors, the decentralized cloud can overcome the limitations raised by centralization. To further explore this topic, we have selected three relevant papers that pave the way for: 1) resource heterogeneity and recommendation in a customer-provided resource cloud platform; 2) price and QoS competition games in cloud service markets; 3) resource sharing model in a federation of selfish cloud providers. Finally, based on the previous work, we discuss our research plan which mainly focuses on the cost and incentive models of the resource allocation.

Index Terms—Decentralized cloud, Resource allocation, Cost model, Incentive model

I. INTRODUCTION

Cloud computing is gaining strength in the IT marketplace because of its on-demand resource provisioning, high availability and elasticity. These features allow cloud end-users to access resources in a pay-as-you-go manner and to meet varying demands without upfront resource commitments. However, in order to accommodate load spikes and to maintain high availability, cloud data centers usually provision more resources than necessary, which results in largely centralized computing infrastructures built by major cloud vendors with rich assets and technologies. Meanwhile, those massive data centers bring about many other problems, e.g., huge energy consumption, bandwidth bottlenecks, privacy, security, legal and physical vulnerabilities. In fact, the expansion of data center is happening at an unprecedented scale and is not sustainable based on the parallel growth of computing and storage requirements. Armbrust et al. [1] show the server utilization of data centers is shockingly low at about 5% to 20%. Data centers are over-provisioning for peak usage sustaining a few days per month, which leads to the underutilization at most of the time.

In the meantime, more and more small data centers are springing up in order to avoid severe issues (e.g., legal [2], privacy, and data control) that can arise with the adoption of massive centralized data centers. For example, in a country like Switzerland with various cantons, one could imagine the emergence of multiple small data centers in various companies and institutions to obviate the reliance on big enterprise data centers (e.g., Amazon EC2, Microsoft Azure, etc.) and to maintain control over local data. Nevertheless, they also confront the same problems of resource over-provisioning, low resource utilization and high maintenance cost. Moreover, due to the limited capacities, those small-scale data centers alone may not be able to provide elastic services with high availability during periods of peak demands. In some cases, they could even act as resource consumers to seek resources from other data centers. Therefore, this motivates small data centers to collaboratively provide cloud services to improve the efficiency of resource allocation.

In this research proposal, we envision a decentralized cloud in which the resources are contributed and shared by a group of small data centers (SDCs). The decentralized cloud is a highly virtualized platform that provides compute, storage, and network services among different SDCs. The power of virtualization makes it possible to manage resources in a virtual execution environment, which facilitates resource trading and sharing. All the SDCs make their decisions on resource allocation in a strategic setting where various strategies based on different preferences (e.g., location proximity, privacy concerns, legal constraints) are employed by SDCs. The strategy
which is used by a participant may depend on what the other
is using and vice versa. We summarize six key characteristics
of the decentralized cloud as follows:

1) **Decentralization.** Each participant independently makes
its own decision on the amount of resources to contribute
based on its demand patterns and strategies. There is no
central authority to account or monitor their behaviors.
Therefore, the participant could also decline the resource
requests from other participants in terms of some strate-
gies or preferences.

2) **Self-organization.** In the decentralized cloud, self-
organization is emerging because of availability, elastic-
ity, legal constraints and location proximity. The partici-
pants could voluntarily establish cooperation networks to
meet their demands.

3) **Geographical distribution.** In contrast with large-scale
data centers which are merely confined to a few locations,
SDCs have a more widely geographical distribution. As a
consequence, decentralized cloud, for instance, will play
an active role in delivering services of low latency due
to location awareness.

4) **Heterogeneity.** Since resources are provided by different
SDCs, they present great heterogeneity in both resource
types (e.g., CPU, memory and network) and demand
patterns (e.g., availability and prices). Such heterogeneity
poses challenges for efficient resource allocation.

5) **Interoperability.** Interoperability between SDCs is of
vital importance for the aggregation of diverse resources
in a decentralized manner.

6) **Privacy and security.** An immediate benefit of decen-
tralization is improved privacy and security since there is
no central control over the data.

Through participating in the decentralized cloud, not only will
small cloud players get a channel to form business groups
and support each other, but also cloud end-users will gain
more confidence in using this model because of new fea-
tures. Indeed, the decentralized cloud combines the advantages
of both cloud computing and Peer-to-Peer (P2P) computing
paradigms. However, there are three key differences be-
 tween the decentralized cloud and P2P model [6]. First, the peers in
our decentralized cloud are mainly small data centers which
are legal and economical social entities rather than individuals.
Thus, the incentives why cloud providers develop cooperation
with each other are not merely for the revenue maximization
but also for other social dynamics such as legal constraints,
location proximity, trust and loyalty. Second, compared to a
large population of peers, the number of cloud providers is
relatively small. Third, the computing environment is not so
highly dynamic as that in P2P model. Compared to continu-
ously joining and leaving of peers in the P2P system, the
participation of SDCs is more stable in decentralized clouds.

Nevertheless, the decentralized cloud is not without prob-
lems. In this research proposal, we will discuss the problems
in the light of reviewing three related papers. The first paper
[9] conducts an empirical study on resource heterogeneity in
a cloud platform built on customer-provided resources. They
also propose a resource recommendation mechanism based on

**BitTorrent swarm application, which bridges the gaps caused
by resource heterogeneity.** Then, we will move to discuss
another big challenge: economic (incentive) models for cloud
markets. The second paper [7] presents an analytical model
for price-QoS competition in cloud services markets based
on non-cooperative game models. In the last paper, Samaan
[8] proposes a self-enforcing resource sharing strategy, which
takes the long-term revenue as a participation incentive, to
promote cooperation in a federation of selfish cloud providers.
Finally, we will discuss our research plan in the Section V.

II. CUSTOMER-PROVIDED RESOURCES FOR CLOUD
COMPUTING

In this paper, Wang et al. [9] investigate a cloud plat-
form built on customer-provided resources: SpotCloud, which
allows customers to sell their idle resources to offer cloud
services collaboratively. In fact, SpotCloud provides a mar-
ketplace for resource **sellers** who sell their idle resources and
**buyers** who buy these resources. They firstly make a mea-
surement study on the characteristics of this platform. Based
on that, they propose an instance recommendation model for
cloud providers to help cloud end-users select instances in
this newborn platform. The proposed recommendation model
converts to a maximum flow minimum cost model, which is
solved by the classical Ford-Fulkerson algorithm.

A. **Resource Heterogeneity**

The resources in SpotCloud are of great heterogeneity as
they are provided by different enterprises and individuals
who have excess resources. The authors present the resource
heterogeneity from two aspects, namely resource capacities
(e.g., CPU, memory and network throughput) and QoS factors
(e.g., geo-distribution, availability, resource provisioning delay
and price). Resource sellers should specify hardware configu-
ratings, location, availability and cost of their resources before
bringing their resources to the market. Resources are traded
in terms of **VM appliances, or instances** which are the same
with the concept of virtual machines.

They investigate 116 instances over one month to under-
stand the basic features of this platform. The instances are
located all over the world (e.g., Australia, Canada, Iceland,
Isle of Man, Italy, Netherlands, Philippines and USA). About
75% of all the instances have less than 4 virtual cores while
80% of them have a memory less than 5GB. However, there
are also some powerful instances with more than 10 virtual
cores and 20GB memory. Regarding the network throughput,
about 50% of them are more than 10 MB.

Unlike cloud servers with high availability, the resource
availability in SpotCloud largely depends on the resource
owners. During the one-month measurement period, the avail-
ability of 40% instances are less than 20%, namely less than 6
days in a month, which makes it more suitable for short-term
running tasks. As for resource provisioning delay, they observe
that most of the instances (about 60%) could be initialized
within 10 minutes while the maximum delay is about 27
minutes. Although the provisioning delay is much higher than
that of public clouds such as Amazon EC2, it is still tolerable
for most of the customers. The price of each instance largely depends on its capacities, and users have diverse options to select instances within their budget.

B. Resource Recommendation

High resource heterogeneity brings new challenges for the adoption of customer-provided resources platform. For example, users have to bear the burden of assessing the performance of highly diverse instances and select the appropriate instances for their applications. To this end, the authors design an instance recommendation mechanism for cloud providers to help users finish instance selections, thereby achieving cost-savings. The target application of instance recommendation is SeedBox which uses a peer-to-peer protocol like BitTorrent for data exchange. They assume that users could report their budgets and tasks to cloud providers who will help the users find a set of instances to maximize the downloading performance within the budget.

In SeedBox application, there are two types of peers: 1) Seeders \(S\) have a complete copy of the file and stay in the system to allow other peers to download from them; 2) Leechers \(L\) only have a part of the file. Moreover, the set of cloud instances selected from the SpotCloud is denoted as \(C\) and each instance \(c \in C\) has a uploading capacity \(u_c\) and a downloading capacity \(d_c\). Given a buyer \(p\) wants to buy a set of cloud instance \(C\) to facilitate his downloading, the performance gains are modeled as the marginal improvements in the completion time of downloading, which is formulated as \(Gain(T, C) = DT(T, p) - DT(T \cup C, p)\). The first part is downloading time without help of cloud instances while the second part refers to the downloading time with cloud instances. The associated cost of downloading with the help of cloud instances is the sum of the cost of renting instances and the cost of data transportation. Thus, given a set of peers \(T = S \cup L\) and a budget \(Q\), a solution of instance recommendation is to find a set of cloud instances \(C\) such that maximize the performance gains and minimize the cost.

In this paper, the authors consider the recommendation optimization as a minimum cost and maximum flow problem which is solved by the classical Ford-Fulkerson algorithm. The authors evaluate their instance recommendation model in both SpotCloud and Amazon EC2. The results show that their model could help users find the best tradeoffs between benefit and cost in both cloud platforms. The cost of using Amazon EC2 instances are generally higher than that of SpotCloud when the completion times of downloading are similar.

C. Discussion

The main contribution of this paper is that they take the first step to study customer-provided resources for cloud computing in which the customers could act as both resource consumers and providers. Their empirical experiments give us many implications for our research. In the decentralized cloud, we also have to take resource heterogeneity into account when designing resource allocation models. At present, cloud end-users bear the burden of translating their performance and cost goals into the corresponding resource requirements [4]. In return, resource providers may suffer revenue loss due to inappropriate resource selection by cloud end-users. Thus, within our decentralized cloud, we should leverage the resource heterogeneity and design different resource recommendation algorithms for various types of popular applications to achieve efficient resource allocations.

However, an important issue which is not touched by the author is that with increasingly more customers’ participation, SpotCloud scheme becomes highly impractical since it requires a large-scale infrastructure for accounting and transactions [3]. Those excessive costs will diminish the price advantage over its competitors like Amazon EC2 and even threaten the survival of the SpotCloud itself. Therefore, business models based on the markets might no longer work, which encourages us to build incentive models based on other factors such as reciprocity, trust and location.

Another interesting future work of SpotCloud is to study the incentives of different customers on why they want to contribute their resources and to utilize others’. Since customers are of high heterogeneous, their incentives may vary greatly. For example, through the SpotCloud platform, some customers could achieve cost-savings while others might seek resources to meet their SLA requirements (e.g., location, response time). Through understanding customers’ incentives, we could design a better business model as well as improve the efficiency of resource allocation.

III. RESOURCE PRICE AND QoS COMPETITION IN CLOUD MARKETS

In this paper, the authors [7] discuss three inter-organizational economical models, which take both price and quality of service (QoS) into consideration, for cloud service markets. Specifically, they consider a monopolistic market with a few big cloud vendors such as Amazon, Google, Microsoft and focus on the non-cooperative game analysis of price and QoS competitions amongst these major vendors.

A. System Model

They consider a market consisting of \(n\) competing cloud providers (CPs) and each CP \(i\) has a price \(pr_i\) and a QoS level \(s_i\) related to a given application type. For simplicity, they only consider the response time and define \(s_i\) as the difference between a benchmark upper bound \(\tau_i\) and the actual response time \(rt_i\). User arrival process is modeled as a M/M/1 queuing system, where user arrival with mean rate \(\lambda_i\) follows a Poisson process whilst service processing with mean rate \(\mu_i\) follows an exponential distribution. This model is based on continuous time Markov chain, which could achieve an expected steady state. Based on the queue theory, we have the average response time is \(rt_i = \frac{1}{\mu_i - \lambda_i}\). It follows immediately that \(\mu_i = \lambda_i + \frac{1}{rt_i}\). From the perspective of cost modeling, this equation could be interpreted as that the service price, \(pr_i\), is the aggregate of a fix cost \(c_i\) caused by user requests (\(\lambda_i\)) and a fix cost \(\rho_i\) raised by resource provisioning (\(\frac{1}{rt_i}\)). Thus, the minimal price is \(pr^{min}_i = c_i + \rho_i, \forall i = 1, ..., n\).
It is intuitive that the user demand should be determined by price and QoS factors. Let $\lambda_i$ as a function of the price vectors $pr_i = (pr_{i1}, \ldots, pr_{in})$ and $s = (s_1, \ldots, s_n)$, and we have demand function as follows,

$$
\lambda_i = \lambda_i(pr_i, s) = x_i(s_i) - y_i pr_i - \sum_{j \neq i} \alpha_{ij}(s_j) + \sum_{j \neq i} \beta_{ij} pr_j
$$

(1)

where $x_i(s_i)$ is an increasing, concave and differentiable function in $s_i$ and function $\alpha_{ij}()$ is increasing and differentiable. Equation 1 is modeled as a separable function of price and QoS vectors, which facilitates the analysis on the independent effects of price and QoS changes over the overall user demand. For example, it is clear that in Equation 1 QoS improvements (or price decrease) by other (or price decrease) by a CP lead to an increase in its user demand whilst QoS improvements (or price decrease) by other competitor CPs result in a decrease in CP $i$’s user demand. Based on demand function $\lambda_i$, the profit function of each CP can be formulated as

$$
P_i(pr, s) = \lambda_i(pr_i - c_i - p_i) - \frac{\rho_i}{r_i - s_i}
$$

(2)

The first part of equation 2 is the revenue from user demand ($\lambda_i$) while the second part is the cost for providing corresponding QoS ($s_i$). In game theoretical analysis, this profit function is called utility/payoff function.

**B. Price-QoS Game**

Given the utility function $P_i$ for each CP $i$ and the strategy space of price $pr_i \in [pr^{min}_i, pr^{max}_i]$ and QoS level $s_i$, the authors study three non-cooperative price-QoS games among multiple cloud providers. In the first game, the authors assume QoS levels are pre-specified and only consider price changes. In Game 2, CPs compete for both price and QoS while in Game 3 price levels are pre-specified and they compete for QoS levels. The solution of a game is to find the Nash Equilibrium (optimal) price and QoS changes for cloud end-users.

In the first game, we notice utility function $P_i$ is supermodular since

$$
\frac{\partial^2 P_i}{\partial pr_i \partial pr_j} = \beta_{ij} \geq 0,
$$

which helps us analyze how one cloud vendor decision affects the incentives of others. The strategy space of price is continuous, $pr_i \in [pr^{min}_i, pr^{max}_i]$. In this context, the supermodularity of utility function $P_i$, implies that an increase in player $i$’s choice $pr_i$, increases the marginal payoff $\frac{\partial P_i}{\partial pr_i}$ of strategy $pr_j$, for all other players $j$. That is, if any player $i$, chooses a higher $pr_i$, all other players $pr_j$, have an incentive to raise their price $pr_j$. The equilibrium price can be obtained by solving the system function of first order conditions $\frac{\partial P_i}{\partial pr_i} = 0, \forall i$.

In the second game, all the CPs compete resources for both price and QoS factors. Thus, the utility $P_i$ is two-variable ($pr_i$, $s_i$) function and the condition of the existence of a Nash Equilibrium (NE) ($pr^*, s^*$) is that $P_i$ should be jointly concave in ($pr_i$, $s_i$). That means the solution NE ($pr^*, s^*$) should satisfy the first-order derivatives $\frac{\partial P_i}{\partial pr_i} = 0$ and $\frac{\partial P_i}{\partial s_i} = 0$. In addition, the determinant of Hessian, $\frac{\partial^2 P_i}{\partial pr_i^2} * \frac{\partial^2 P_i}{\partial s_i^2} - (\frac{\partial^2 P_i}{\partial pr_i \partial s_i})^2$, should be large than 0. Thus, multiple Nash equilibria might exist with holding both conditions. When CP price is specified, the QoS can be computed immediately. In other words, Game 3 is a special case of the Game 2.

**C. Discussion**

This paper presents an analytical model for cloud service markets and formulates non-cooperative price and QoS games among multiple cloud providers. Although the authors do not conduct any experiment or simulation based on their models, they suggest that it might take a long time for the convergence of NE. Such models, however, can quickly become intractable due to the increasing computational complexity. Thus, we may have questions: Is game theory the right tool? What characteristics do these models capture and what are missing?

Based on this paper, their models only reveal that how price and QoS changes influence the profits of cloud providers whilst there are still many other factors to consider such as location proximity, legal constraints. Furthermore, we can model the cloud markets with a multiagent environment to simulate a more fleshed out cloud scenario.

Another shortcoming is that they only consider how to derive the optimal price and QoS levels to maximize individual CP revenues under the competitions. However, there are also a lot of cooperation among competitors. In our decentralized cloud scenarios, we will pay more attention to the factors which promote cooperation among small cloud vendors. By understanding what factors and conditions are necessary for cooperation to emerge, appropriate actions can be taken to foster the development of cooperation.

**IV. RESOURCE SHARING MODEL IN CLOUD FEDERATION**

In her work [8], Nancy presents an economical model for resource sharing in a federation of selfish cloud providers (CPs). All the CPs in the federation can share their unused resources during the periods of low-demands and borrow spare capacities during the peaks to maximize their revenues. The model proposed in this paper not only considers the instantaneous revenue but the CP’s long-term revenue as well.

**A. System Model**

They consider a federation $N$ of $N$ cloud providers, $CP_1, \ldots, CP_N$. Each of them has a capacity $C_i$ which denotes the number of VM hosted by $CP_i$. Assume a discrete time horizon, $t = 1, 2, \ldots$, where at the beginning of time $t$, $CP_i$ observes the total demand $d_i(t)$. In other words, $CP_i$ has idle resources, $e_i(t) = C_i - d_i(t)$. The problem is that how $CP_i$ determines the number of VMs to offer in the spot market, $w_i(t)$, and that to offer for sharing in the federation, $e_i(t) - w_i(t)$. Offering resources for federation is free of charge while the revenue obtained from the spot market is increasing, concave and twice continuously differentiable function, $r_i(w_i(t)) = \frac{dr_i(w_i(t))}{dw_i(t)}$, is the additional revenue generated by the additional resources.
1) Markovian model for the workloads: Markovian model is proposed to model the transitions of the CPs’ idle resources. At each time \( t = 1,2,\ldots \), the observed unused capacity of the \( N \) CPs is denoted as \( s_t = \{e_1(t),\ldots,e_N(t)\} \), which follows a Markov process. \( s_t \) is from a finite set of state \( S = \{s_1,\ldots,s_T\} \) and a state transition matrix is defined as \( \Pi = [\pi(s_i|s_j)]_{i,j = 1,\ldots,T} \). A history of state observation is denoted as \( h_t = (s_1,s_2,\ldots,s_t) \). \( w_i(s_t) \) and \( w_i(h_t) \) are the allocation of spot market in the state \( t \) and history, respectively. Due to the Markov property, unused capacities at \( s_t \) are history dependent, i.e., \( e_i(h_t) = e_i(s_t) \), \( \forall i \).

2) Limitation of non- and fully-federated models: We firstly discuss the problem from a perspective of non-federated model in which the individual CP aims at maximizing its revenue \( U(w_i(h_t)) = \sum_{h_t=1}^{\infty} \delta^t \sum_{h_t} \pi(h_t) r_i(w_i(h_t)) \). Since \( r_i(.) \) is monotonically increasing, it immediately follows that each individual CP should allocate all their idle resources to the spot market, i.e., \( w_i(h_t) = e_i(s_t) \), to maximize their revenue. Thus, their revenues largely depend on their current demand patterns of the guaranteed-service users. Nevertheless, due to lack of federation, the individual CP loses the chance to increase their revenue through borrowing resources from other CPs, which results in the loss of the long-term revenue.

On the other hand, we may think of a federation as a society of \( N \) CPs. While each CP makes their local decisions on how many resources to share, we can also analyze the system from a global or societal point of view and may find out certain allocations of resources over others. Within the society, all the unused resources are managed by a centralized resource broker. The goal of resource broker is to find out a socially optimal allocation which maximizes the society revenue. The broker attaches non-negative weights \( \lambda_i \), \( i = 1,\ldots,I \) on the individual CPs and chooses allocations \( w_i(h_t) \) to maximize
\[
W = \sum_{i=1}^{N} \lambda_i U(w_i(h_t))
\]
It follows that if all unused resources are allocated in the spot markets, i.e., \( \sum_{i=1}^{N} w_i(h_t) = \sum_{i=1}^{N} e_i(s_t) \), the revenue of federation achieves the maximum. However, this arises the problem that the marginal revenue of \( CP_i \) at each time \( t \) depends on the weight \( \lambda_i \) rather than the number of unused resources of current state. For example, if all the CPs share the same function \( r(.) \) and weights \( \lambda \), the amount of unused resources will be distributed equally on their markets. There is no doubt that the socially optimal allocation guarantees the maximum revenue for the federation. However, the distribution of revenue generated by this allocation is determined by exogenous constant weight \( \lambda_i \) assigned by resource broker rather than the real contributions of each individual CP. Hence, rational CPs aiming at maximizing their own profit cannot be enforced to cooperate and share their resource in the federation. In other words, their cooperation is not stable.

3) A Game-theoretic Framework: A general game theoretic framework is proposed to represent two extreme models. A stage game is defined as a triplet consisting of a set of players, strategies available for each player and payoff functions for each strategy. In a federation, all CPs participate in the game < \( N \), \((w_i(s_t))_{t=1}^{N} \), \((r_i(w_i(s_t)))_{t=1}^{N} \) > by determining the best resource allocation to maximize their revenues. The solution of a game is a Nash Equilibrium (NE) which no single player has an incentive to change their chosen strategy after considering all the other players’ strategies. In the repeated game setting, CPs are interested in finding a subgame-perfect Nash Equilibrium (SPNE) which is a NE for CPs in every subgame (i.e., up to any history \( h_t \)). Apparently, the solution to non-federated model is the NE for this game since no CP gains from lending its resources. However, this paper tries to find a better SPNE that overcomes the long-term revenue loss in non-federated model and enforcement limitation in that of fully-federated model.

In other words, the proposed approach in this paper should not only guarantee that each CP’s revenue is at least equal to its non-federated revenue but also ensure the self-enforcement of these strategies. Basically, the idea is originated from risk sharing in economic model. At each time \( t \), CPs could share part of their excess resources for the federation to build a model of informal insurance against the risk of future peaks in the workloads. The long term revenue of a CP who refuses to share spare resources with others will be reduced to that in non-federated model since other CPs will no longer lend their resources to a selfish CP. Thus, the proposed approach uses the uncertainty in future revenue as a participation incentive, which is mathematically formulated as follows.

\[
P1 : \max_{w_i(h_t)} \sum_{i=1}^{N} \lambda_i \sum_{t=1}^{\infty} \delta^t \pi(h_t) r_i(w_i(h_t)) \tag{3}
\]
subject to
\[
\sum_{i=1}^{N} w_i(h_t) \leq \sum_{i=1}^{N} e_i(s_t) \tag{4}
\]
\[
r_i(w_i(s_t)) + \sum_{\tau=t+1}^{\infty} \sum_{h_{\tau}} \delta^{\tau-t} \pi(h_{\tau}|h_t) r_i(w_{\tau}(h_{\tau})) \geq R_i(s_t) \tag{5}
\]
where \( \forall h_t, \forall \delta \in (0,1) \) is the discount factor and \( \lambda_i \) is the normalized exogenous constant weight such that \( \sum_{i=1}^{N} \lambda_i = 1 \). The constraint (4) is the capacity constraint while (5) is defined as the commitment constraint which guarantees the long term revenue of a CP at least equal to that in non-federated model, i.e., \( R_i(s_t) \).

B. Recursive Formulation

To solve the problem, the author firstly analyzes the problem with Lagrangian multiplier
\[
\sum_{t=1}^{\infty} \sum_{h_t} \delta_t \pi(h_t) \sum_{i=1}^{N} \lambda_i r_i(w_i(h_t)) + \zeta(h_t) \sum_{i=1}^{N} (w_i(h_t) - e_i(s_t)) + \eta_i(h_t) [\sum_{\tau=t}^{\infty} \sum_{h_{\tau}} \delta^{\tau-t} \pi(h_{\tau}|s_t) r_i(w_{\tau}(h_{\tau})) - R_i(s_t)]
\]
where \( \zeta(h_t) \) and \( \eta_i(h_t) \) denote the multipliers for constraints (4) and (5), respectively. The first order condition of Lagrangian with regard to \( w_i(h_t) \) yields the ratio between the
CPs’ marginal revenue:

\[
\frac{r_j(w_i(h_t))}{r_j(w_j(h_t))} = \frac{\lambda_j + \sum_{t=1}^{t} \frac{\eta_j(h_t)}{\beta_j(h_t)}}{\lambda_i + \sum_{t=1}^{t} \frac{\eta_i(h_t)}{\beta_i(h_t)}} = \frac{\lambda_j(h_t)}{\lambda_i(h_t)}
\]  
(6)

It is clear that the ratio of marginal revenues between any two CPs is determined by the initial weights and a history of multipliers \(\eta_j(h_t)\). The difficulty of this problem is that the dimension of histories \(h_t\) increases exponentially over time. However, due to Markov properties of the workload, the problem can be cast as a recursive one with Bellman equation as follows. We rewrite the revenue function \(r_i(w_i(h_t)) = r_i(w_i(\lambda^t_{i-1}), s_t)\) and the problem 1 could be cast as follows,

\[
R_i(\lambda^t_{i-1}, s_t) = \max_{w_i} \sum_{s_{t+1}} \delta \pi^{s_{t+1}}_i R_i(\lambda^t_{i}, s_{t+1})
\]  
(7)

with the commitment constraint,

\[
r_i(\lambda^t_{i-1}, s_t) + \delta \sum_{s_{t+1}} \pi^{s_{t+1}}_i R_i(\lambda^t_{i}, s_{t+1}) \geq R_i(s_t)
\]  
(8)

The problem could be solved using iterative methods to obtain the optimal values of \(w_i(.)\) and \(\lambda^t_i\). The paper gives updating rules for \(\lambda^t_i\) based on the analysis of lower and upper bounds.

The results of their simulation show that when the CPs have enough patient with \(\delta = 0.98\), the revenue created by proposed model is almost the same with that of full-federated model. As \(\delta\) decreases, the revenue of federation decreases but is higher than that of non-federated model. Meanwhile, the proposed model smooths the fluctuations in the workloads greatly because of the large variance of VM availability.

C. Discussion

Compared to [7], this paper models the interactions among cloud providers as a repeated game. They consider the long-term revenues of CPs which could act as a participation incentive to share unused resources. In every round of the repeated game, each CP should determine the amount of resources which will be shared with others. Thus, the performance of their model largely depends on the rationality of each CP. However, in the real world, we should consider cloud providers with bounded rationality. A repeated game among multiple CPs with bounded rationality is more like a learning process or an evolution of cooperation. During each stage of game, every CP learns how to cooperate with each other and finds its optimal strategy during the evolution. We can simulate a decentralized cloud with different types of CPs, ranging from rational to bounded-rational ones. Through this, we could observe how these different strategies affect the efficiency of resource allocation and how stable the cooperation is.

One common problem in both papers [8], [7] is that they only focus on the revenue maximization. However, in decentralized clouds, the incentive why small data centers cooperate with each other is not limited to the profit growth. For example, with more interactions, two collaborators may generate high loyalty through reinforcement learning [5], which also has a great effect on their preference of resource allocation.

Last but not least, the author models workload demands based on Markov process in which CPs could make predictions of the future workload based solely on the present state. It would be interesting to explore the impact of different workload demands (e.g., heavy-tailed, Poisson arrival) on the resource allocation.

V. Research Proposal

In this section, we will discuss our research plan from different dimensions of resource allocation in decentralized clouds. An examination of the relevant literature, some of which we discussed above, reveals the following potential directions:

A. Utility Maximization Model based on Bounded Rationality

In the literature, the economic ideas that have been employed for resource allocation and incentive models, in the decentralized cloud scenario, are imbued in the rational utility optimization framework. Utility maximization with full rationality is a useful idealization that has been employed by economists to analyze complex situations. However, it has often been argued that the assumptions entailed in this framework often do not hold in real life scenarios [3].

Unlike complex models with strict rationality in previous work, we shall apply a utility maximization model to a more fleshed out cloud scenario based on bounded rationality. In order to model decentralized cloud scenario more realistically, we shall explore more dimensions which have an effect on resource allocation. To begin with, we will incorporate social dynamics such as trade barriers, loyalty, trust among the CPs and present an analytical model in the decentralized cloud scenario. Next, we could add in other factors such as biased preferences of different CPs, legal constraints, geographical proximity, infinite and finite time horizon, etc.

B. Multiagent Resource Allocation

Utility maximization models usually require high levels of abstraction and thus cannot model complex strategies easily. In fact, such models can quickly become intractable with increased complexity [8],[3]. Thus, we can use a multiagent simulation to model resource allocation strategies of different cloud providers, including those based on reciprocity, loyalty, location proximity, legal constraints etc.

For example, we can simulate a decentralized cloud with different types of cloud providers (e.g., rational, bounded rational, obedient and malicious), having goals more than utility maximization. To evaluate the impact of those different strategies adopted by cloud vendors, we could design the metrics from two perspectives. From the perspective of cloud providers, they are more interested in observing the impact of different strategies on data center performance such as resource utilization, cost and availability. From the strategy’s point of view, we would like to evaluate each strategy in terms of robustness, stability and fairness. For example, we can design two groups of possible experiments. In the first group of experiments, we simulate a population of CPs with different
resource allocation strategies to find out which strategy does better in terms of some metrics like resource utilization and availability. The second group of experiments are to evaluate the performance of the optimal strategies. For instance, we could evaluate the stability of a strategy from an attacker’s point of view and observe whether a strategy can resist invasion by attacker’s strategies.

C. Business Cluster based on Coalition

A very important factor that has not been studied extensively is the formation of Business Clusters among small cloud providers based on geographical and legal proximity. Paul Krugman, the Nobel Prize winning economist, has emphasized that individual businesses often combine to help each other due to geographical proximity, legal proximity and other factors. Although a federation based model has been considered in the last paper [8], a fleshed out model of individual cloud providers entering into business clusters based on diverse factors such as trust, reputation and in-group bias (e.g., data centers within Switzerland would like to form clusters with each other) has not been analyzed.

Using both a cooperative game-theoretical approach and a multiagent system approach, we shall model the interactions of cloud providers in terms of coalition. The coalition formation problem can be divided into two parts: selection of the coalition members and sharing the value created by the coalition among its members. Thus, one of the central points is the stability of the coalition. One possible experiment is to simulate a group of cloud providers with different strategies (e.g. based on location proximity, loyalty, legal constraints). The idea is to observe which strategies result in more stable coalition. On the other hand, if there is no stable coalition to emerge, we should design a utility transfer strategy which could redistribute the value created by the coalition, such that no cloud vendor has an incentive to leave the coalition.

D. Resource Allocation Graph/Network

In the decentralized cloud, all the participants contribute different amount of resources, which forms a resource network. Hence, we can formulate resource allocation in a network as a weighted graph game. The vertexes represent cloud providers while the weights are resource allocation preferences of CPs. In addition, we can define characteristic function to map any coalition of cloud providers (i.e., any subset of vertexes in graph) to a value, indicating the total value these cloud providers achieve together. Then the problem is to find the optimal coalitions within the graph. It is evident that computation complexity of this problem is hard but we can find good approximate solutions.

E. Other Dimensions

In the long-run, we could also explore other dimensions of resource allocation in the decentralized cloud. For example, the availability of resources provided by different vendors exhibits strong temporal and spatial correlations, using which we can design a resource plan to ensure high service availability. Furthermore, due to the limited capacities of small data centers, it is interesting to design a resource scheduler to relocate the workloads among data centers, thereby improving the elasticity of small data centers.

REFERENCES