On Incentive of Customer-Provided Resource Sharing in Cloud

Haiyang Wang
University of Minnesota at Duluth
Minnesota, USA
Email: haiyang@d.umn.edu

Jiangchuan Liu
Simon Fraser University
Burnaby, Canada
Email: jcliu@cs.sfu.ca

Ke Xu
Tsinghua University
Beijing, China
Email: xuke@tsinghua.edu.cn

***Abstract*—The state-of-the-art cloud computing service has attracted significant interests from the Internet users. However, in the existing cloud platforms, the cloud users are pure consumers; their local resources, though abundant, have been largely ignored.**

In this paper, we for the first time explore the resource pricing as well as the incentive issues in SpotCloud, a real-world system that enables customer-provided cloud computing service on the Internet. In this system, the resource providers are largely heterogeneous and are not forced to contribute their resources. A working business model is therefore important to offer them enough sharing incentive. Instead of setting a standardized pricing rule for unit resource, we suggest a distributed market that allows the sellers to decide the quality, quantity, and pricing of their own resources. We demonstrate the efficiency of this business model through a repeated seller competition game. The trace-analysis further indicates that the proposed business model can successfully motivate the resource sharing in our Spotcloud system.

I. Introduction

Over the past decade, cloud computing has become one of the defining development trends of the Internet applications. The cloud service providers, such as Amazon [1] and Rackspace [2], have also experienced a great leap forward in terms of both system capacity and user popularity. Besides this traditional datacenter-based cloud service, there is now an increasing interests to utilize customers' local computing resources for cloud computing [3]. Such industry leaders as Enomaly[1] also take their first steps towards the feasibility and the system design of enabling customer-provided resources for cloud computing. This new generation of cloud system, beyond conventional datacenter-based design, enables a cost-effective and flexible alterative to complement the existing enterprise cloud. However, given that their cloud resources are provided by Internet users, how to enable the incentive to motivate their resource sharing remains a critical issue in the system design.

In this paper, we for the first time investgate the resource pricing and the incentive issues in `SpotCloud`[2], the first and the largest customer-provided cloud system on the Internet. Since the potential resource providers (*sellers*) in SpotCloud are heterogeneous and are not forced to contribute their resources, a working business model is necessary to offer them enough incentive. Instead of setting a standardized pricing rule for unit resource, we suggest a distributed market that allows the sellers to decide the quality, quantity, and pricing of their local resources to be contributed. We further demonstrate the efficiency of this business model through a repeated seller competition game. Our analysis indicates that in this system, the price per unit resource can be converged to a stationary equilibrium. Moreover, the sellers will not have the incentive to carry out such harmful strategies as pricing monopoly and cooperative cheating.

The rest of this paper is organized as follows: In Section II, we present the related works. Section III gives an overview of SpotCloud system and Section IV examines the resource pricing problem. We further discuss some piratical issues in Section V and Section VI concludes the paper.

[1]http://www.enomaly.com/
[2]http://www.spotcloud.com/

II. Related Works

Cloud computing is reshaping the landscape of the Internet applications and provides countless new opportunities for both industry and academia [4][5][6][7]. There have been a series of measurement and comparison of the diverse cloud platforms. Garfinkel *et al.* studied the performance of Amazon's Simple Storage Service (S3) and described the experience in migrating an application from dedicated hardware to S3 [8]. Walker *et al.* investigated the performance of scientific applications on Amazon EC2 using both macro- and micro-benchmarks [9]. A recent study by Li *et al.* [10] further presented CloudCmp, a systematic comparator for the performance and cost of cloud providers in today's market. These studies have mainly focused on cloud enabled by enterprise datacenters. They have demonstrated the power and benefit of such enterprise clouds, but also revealed many of their weaknesses. In particular, Ward [11] showed that the virtualization technique in Amazon EC2 can lead to dramatic instabilities in network throughput and latency, even if the datacenter network is only lightly loaded. To rebalance the workloads across physical machines, VM (virtual machine) migration is also widely suggested [12]. A recent study by Shrivastava*et al.* [13]introduced an

978-1-4799-2003-7/14/$31.00 ©2014 IEEE

application-aware migration approach which can greatly reduce the network traffic during the VM migration.

Our work is related to peer-to-peer, which also seeks to utilize local user resources. Yet most of the peer-to-peer systems have focused exclusively on content sharing [14], while a cloud platform is expected to offer diverse resources for a broad spectrum of services. Moreover, though certain incentive mechanisms have been developed for peer-to-peer systems to penalize free-riders, there still lacks clear business/pricing models to enable enterprise-level services, not to mention the economic incentive as we are targeting.

III. Enabling Customer Resources for Cloud

A. Framework Design

In Figure 1, we outline the relation between the cloud providers and their customers. The solid lines illustrate their relations in the existing cloud, with the customers being pure resource-buyers. As such, their local resources have been largely ignored or exclusively used for each individual's local tasks, which are known to be ineffective. Aiming at mitigating this gap between centralized data-center resources and the distributed local resources, our framework enables individual cloud customers to contribute their otherwise exclusively owned (and idled) resources to the cloud and utilize others' resource if needed, as illustrated by the dotted lines in the figure.

B. SpotCloud: A Practical System Implementation

We now present a real-world system implementation, namely Enomaly's `SpotCloud`. As shown in Figure 2, SpotCloud consists of three key modules: Cloud management, Account management, and User interface. The cloud management module supports a variety of common *hypervisors* (also known as *virtual machine managers*) including Xen, KVM and VMware as well as a highly fault tolerant and distributed *Extensible Messaging and Presence Protocol* (XMPP) with built-in failover capabilities. It also works with our resource provisioning algorithm for VM provision and migration. The account management, built on the Google App engine, allows the customers to create Google accounts for the SpotCloud marketplace. This marketplace is provided by the user interface module to let the potential sellers post and update their configurations and prices for the contributed resources.

Figure 3 shows a simplified finite-state machine (FSM) in the SpotCloud system, where the *Authentication* state is managed by the account management module; the *Wait*, *Open* and *Billing* states are managed by the user interface module, and the rest of states are managed by cloud management module. The dark circles refer to the states that are used to communicate with the sellers. In particular, SpotCloud uses a set of *RESTful* (wait for sellers' information/reply to go to the next state) HTTP-based APIs for such communications. Figure 4 shows an example of the message format when SpotCloud sends a *HTTP utilization monitor request* to a seller, where *loadfifteen* field includes

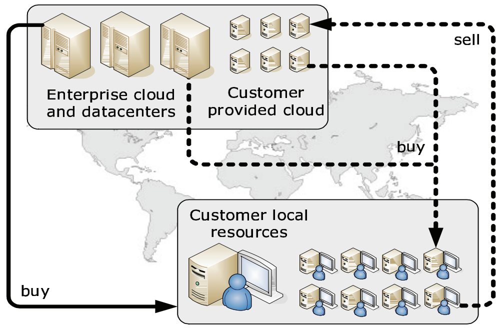

Fig. 1: Overview of Framework

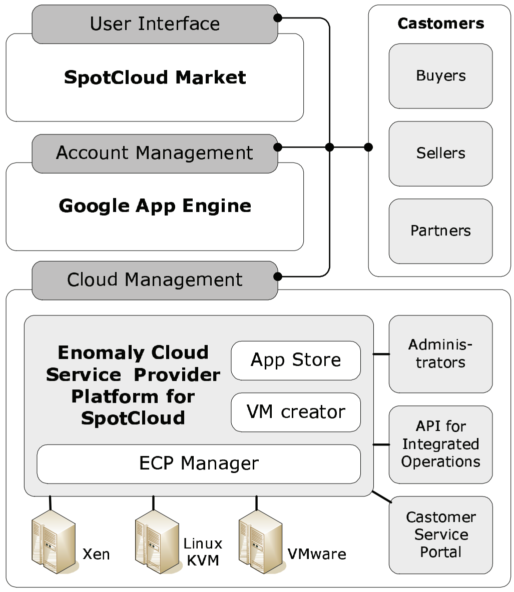

Fig. 2: Software module design

the average load over the past fifteen minutes for the seller. More details can be found in our API and Third Party Provider Integration Guide [15].

It is easy to see that most critical challenge in Spotcloud is to offer enough incentive for a customer to contribute her/his resources or utilize others'. This problem is further complicated given that the customers are highly heterogeneous, making the coarse-grained pricing model used by the existing cloud providers hardly working. To mitigate this problem, we will discuss the design of our market model in the next Section.

IV. Pricing with Heterogeneous Resource Providers

In most of the existing enterprise cloud platforms, fixed pricing remains the most popular strategy. Amazon EC2, as a typical example, advertises $0.02 − 2.62 per hour for each of its *On Demand Virtual Machine* instances, depending on their types. Recently, dynamic pricing also been introduced, e.g., the "spot pricing" in EC2 [16] that aims at better

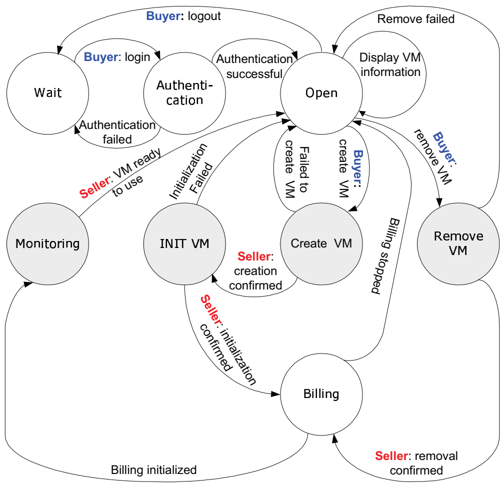

Fig. 3: Finite-state machine in SpotCloud

Request sent by SpotCloud:

https://api.provider.com/utilization?
login=Login
&ecp_username=39480304
&ecp_auth_digest=
lfOBcOAfcLPqPUz1b1dE4MYQFSw=

Response returned by resource sellers:

```
{
  total_memory: 4085,
  free_storage: 84146,
  free_memory: 1397,
  total_storage: 291618,
  loadfifteen: 1.7
}
```

Fig. 4: An example of message format for utilization monitoring

TABLE I: List of notations

N	Total number of sellers
I	The set of VM sellers where each seller $i \in I$
x	Vector of buyers' demands on sellers
c	Vector of sellers' capacities
p	Vector of sellers' prices
q	The quantity actually sold capacities
γ	Vector of sellers' cost factors
$\bar{S}$	Strategy set a cross N sellers i
S_i	The set of strategies for seller i
t	Time slots
$Z(t)$	Strategy vector at time t
D	Amount of user demand faced by seller
δ	Discount factor

utilizing the vacant capacities in the datacenters. It is known that the spot price will be dynamically adjusted to matching the supply and demand, though the full details have not been disclosed.

Since the potential resource providers in SpotCloud are heterogeneous and are not forced to contribute their resources, a working business model is necessary to offer them enough incentive. Therefore, instead of setting a standardized pricing rule for unit resource, we suggest a distributed market that allows the potential providers (i.e., *sellers*) to decide the quality, quantity, and pricing of their local resources to be contributed, in which:

(1) A seller will advertise the configuration (amount and availability) of its local resources as well as the asking price; such information will be seen by other sellers and buyers;

(2) Both resource sellers and buyers are rational: given the advertised prices, a buyer will try to minimize the cost for resource provisioning, and a seller will try to maximize the profit;

(3) After seeing others' advertised information, a seller will adjust her/his own configuration and price to maximize her/his potential profit.

The intuition behind this design is that the sellers have better knowledge of their own resources in terms of both running costs and expected values. If they cannot find a good way to gain profits, any fixed or dynamic pricing rule will fail to give them the incentive to join cloud markets. This business model can be formulated as a variation of a *Repeated Seller Competition* game [17] as follows. To facilitate our discussion, we list the key notations that will be used in this paper in Table. 2.

We start with a general model with N resource sellers. Each seller can be thought of as a cloud service provider operating s/he own local resources. We denote the vector of sellers' resource by $c = (c_1, ..., c_N)$. Investing in resource is costly. In particular, the cost of sharing resource c_i for seller i is $\gamma_i c_i$ where $\gamma_i > 0$ for $i \in \{1, ..., N\}$. We denote the price charged by seller i (per unit demand) by e_i and denote the vector of prices by $e = (e_1, ..., e_N)$. On the other hand, the buyer demand is given by a set $W = \{w_1, w_2, ...\}$, which may result from a continuum of consumers who demand exactly one unit up to their *reservation prices*(the unit price that expected by the buyers). Therefore, $|W|$ shows the aggregate demand in the system.

For the sellers in the system, their possible strategies are: (1)Provides c_i cloud resources with unit price e_i; (2) Inactive and not willing to contribute any local resources (n.a). This strategy space can be written as:

$$S_i := \{e_i, c_i\} \bigcup \{n.a\} \qquad (1)$$

where $\{e_i, c_i\}$ means that seller i is active and provides c_i cloud resources with unit price e_i, and $\{n.a\}$ means that seller i is not willing to contribute any local resources. The game is thus played as follows: each seller i announces a capacity c_i and a unit price e_i which it is willing to sell. In this case, its profit is given by:

$$\pi(S_i) = \begin{cases} e_i \cdot c_i - \gamma_i \cdot c_i - F & for \quad e_i \in [0, v] \\ 0 & else \quad when \quad S_i = n.a \end{cases} \tag{2}$$

where q_i (which will be determined precisely below) is the resource capacity sold by seller i, and F is the fixed cost (e.g. the deployment overhead) when sellers configure their local resources for cloud service[3]. v is the *choke off price* indicating the maximin price of the sellers; no demand will face to seller i if $e_i > v$. The objective of each seller is to maximize the profit π.

Note that, all buyers will place their orders with the cheapest sellers. These orders are fulfilled until the cheapest sellers' capacities are all exhausted. If not all orders can be fulfilled by the cheapest sellers, we assume that the buyers will be rationed based on the Beckmann-rationing rule [18]. In particular, a random sample of consumers will be served by the cheapest sellers. The remaining (unserved) buyers will place their orders with the next cheapest sellers. This procedure is repeated until either all the customers are served or all sellers are exhausted. The demand faced by a seller i is given by:

$$H_i = \frac{max\left\{|W| \cdot (1 - \sum_{j:e_j<e_i} \frac{c_j}{|W|_{[w=e_j]}}), 0\right\}}{\sum_j I_{[e_i=e_j]}} \tag{3}$$

where $|W|_{[w=e_j]}$ is the number of demands with reservation prices equal to e_j and $I_{[e_j=e_i]}$ is the number sellers with the prices equal to e_i. Thus we have $q_i = min\{H_i, c_i\}$. Since it is a repeated game, when we use $Z(t)$ to denote the strategy vector of period t , the expected payoff of seller i over all infinitely many periods is given by:

$$\sum_{t=1}^{\infty} \delta \pi_i[Z(t)] \tag{4}$$

where $0 < \delta < 1$ is a *discount factor*, denoting the sellers' patience (close to 0: not patient; close to 1: highly patient). In practice it denotes the probability of whether the sellers will have the chance to play the game in the future [19]. According to Folk Theorems [20], the optimal method of playing this repeated game is not to repeatedly play a Nash strategy of the stage games, but to cooperate and play a socially optimum strategy. We will thus discuss whether such a stationary outcome equilibrium exist in this repeated game.

[3]Notice that if there are no positive fixed costs, selling nothing would yield zero profits. So the difference between activity and non-activity vanishes in that case, and the extension of the strategy space by the element "n.a." would not be necessary. More importantly, the configuration of local resources involves overheads in real-world; it is thus necessary to consider it in our model.

Assume that $\bar{S}$ is a stationary outcome equilibrium set across N sellers in the repeated game with unit price e (where the sellers charge the same price per unit resource in each period). Since all sellers are active and the game is symmetric, we omit the index i where no confusion results. We further write the sellers' profit in this stationary equilibrium as follows:

$$\pi(\bar{S}) = \begin{cases} (e - \gamma_i) \cdot \frac{|W|_{[w \le e]}}{N} - F \quad if \quad \frac{|W|_{[w \le e])}}{N} < c \\ \\ (e - \gamma_i) \cdot c - F \qquad\qquad else \end{cases} \tag{5}$$

Since δ is a probability, we have $1 + \delta^2 + \delta^3 + ... = \frac{1}{1-\delta}$. Thus the expected profit (over infinitely many periods) at this stationary equilibrium is $\frac{1}{1-\delta}\pi(\bar{S})$. It is easy to see that the stationary outcome equilibrium will exist when $\frac{1}{1-\delta}\pi(\bar{S}) > 0$ Otherwise if $\frac{1}{1-\delta}\pi(\bar{S}) \le 0$, the sellers will prefer to be inactive ($n.a$) with zero profit. In other words, if there is a strategy set a cross N sellers such that $\pi(\bar{S})$ is positive, the stationary outcome equilibrium will always exist for δ sufficiently close to 1. Since the lower part if Eq.6 is the upper bound of $\pi(\bar{S})$, it is not difficult to see that $\pi(\bar{S})$ is decreasing in N; it will further becomes negative for N sufficiently large due to the existence of F. Hence, we can give the upper bound of N in the system (the existence condition for the stationary equilibrium) as follows:

$$N^* = max\{N|\pi(\bar{S}) > 0\} \tag{6}$$

Therefore, for all $2 \le N \le N^*$ there is always a stationary outcome equilibrium $\bar{S}$ for the system where the sellers charge the same price per unit resource in each period. An intuitive explanation for N^* is the maximum number of sellers in the cloud market. Moreover, if $\bar{S}$ is a stationary outcome equilibrium, it is easy to see that for any seller i $S_i \neq n.a$ (since the profit is larger than zero, the sellers will have no incentive to be inactive). Note that the unit price e for this stationary outcome equilibrium is not unique, and it is depending on the buyers' demand at price e: $L(e) = |W|_{[w \le e]}$. The bound of this price will be further discussed in our future works.

We can learn that if we deploy a cloud market where the sellers can dynamically adjust their selling quantity and prices, the sellers will be able find suitable strategies to sell their resources for a higher benefit (instead of leaving the market which results to zero profit). In this business model, the sellers' incentive and their total population are both depending on the buyer demand. Large demands can give sharing incentive to more sellers (larger N^*) and smaller demands will reduce the number of N^*. Yet, unless $|W| = 0$, this business model can always give incentives to a given number of sellers to join the cloud market. Moreover, giving the infinite round of this game, the price per unit resource has the trend to converge to a stationary equilibrium [21].

Different with the fixed pricing models, this stationary price is dynamically decided and adjusted by the sellers based on buyers' demand.

It is also worth noting that the pricing monopoly and cooperative cheating can hardly happen in such a system. The main reason is that there are also some other alternative cloud providers over the Internet such as EC2 and Google. If the sellers' prices exceed the unit prices in other platforms, the sellers' profits will also be decreased to zero. This feature was also captured by the choke off price v in Eq.4.

A. Measurement-Based Validation

Based on this business model, SpotCloud platform has already attracted the Internet customers worldwide. As we have discussed in the previous subsection, the price of SpotCloud VMs is customized by individual sellers who provide/sell their cloud capacities. As shown in Figure 5, we can see that the SpotCloud VMs are mostly very cheap. Moreover, this curve is also quite smooth indicating that the buyers have very high flexibility in selecting VMs in this customer-provided cloud platform. It is also worth noting that the unit price of SpotCloud resources are quite stable overtime. In particular, during our 30-day measurement, only 5 sellers adjusted their unit prices. The changes are within 0.1USD/hour which validates our model analysis in Section IV.

We further check the number of contributed computational capacities in the SpotCloud VMs. As shown in Figure 6, it is easy to see that most SpotCloud resources ($> 75\%$) possess less than 4 virtual cores. Each virtual core provides the equivalent CPU capacity of a $1.0 - 1.2$ GHz 2007 Opteron processor. This is not surprising since most of the customer-provided resources are not as powerful as those from enterprise datacenters. Yet, there are also some relatively powerful VMs; for example, a customer-provided VM has 16 virtual cores with 2 computation units in each core, which is capable of running certain CPU intensive tasks. We also show the memory sizes on the VMs in Figure 7. We can see that most VMs (80%) in SpotCloud have a memory less than 5GB, which is not extra huge but is suitable to run most of the real-world tasks. It is worth noting that, the curves of SpotCloud VMs are quite smooth, indicating the existence of more flexible options to meet the heterogeneous demands from customers. We can see that the proposed business model successfully motivates the Spotcloud sellers to post their local resources in the marketplace. This gives other users more flexible choices, in terms of both performance and pricing, to utilize the cloud resources.

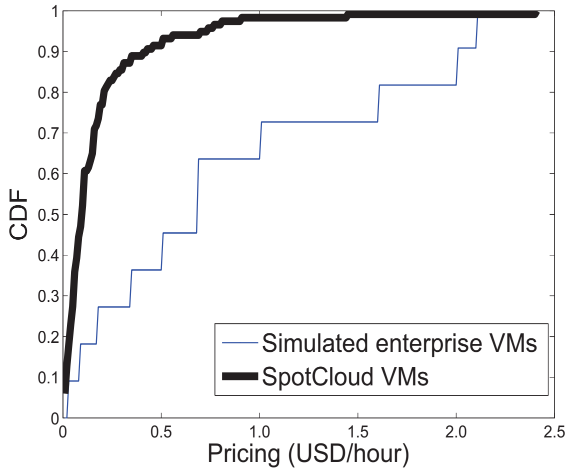

Fig. 5: Pricing distribution

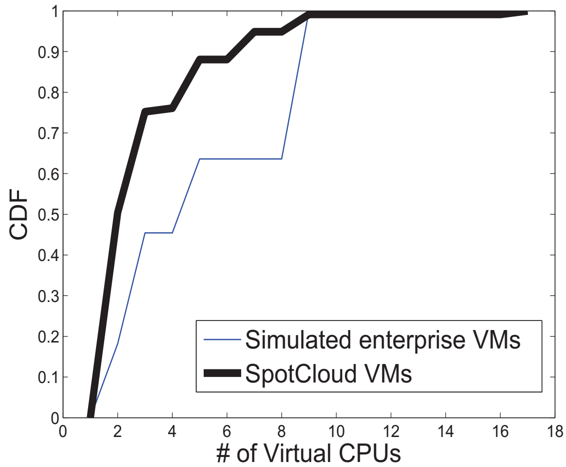

Fig. 6: # of CPUs

V. Further Discussion

Our work represents an initial attempt towards the incentive of Spotcloud-like cloud systems. There are many possible future avenues. Besides the incentive problems, we are particularly interested in the following three three issues:

Privacy and Security: It is worth noting that the utilization of customer-provided resources can introduce privacy, security and authorization issues [22] to the cloud systems. Similar to the peer-assisted content storage systems, we believe that such a problem also needs to be carefully mitigated in our system. Not to mention the existing encryption functions in Spotcloud, one of our ongoing works is to design a privacy-aware load assignment for such a customer-provided cloud platform.

Communication Latency Between VMs: Spotcloud is a highly distributed cloud platform. This feature largely prevents the single point of failure while also introduces extra communication latency between VMs. To mitigate such a problem, the customers need to carefully address the locality-awareness issue and reduce network traffic

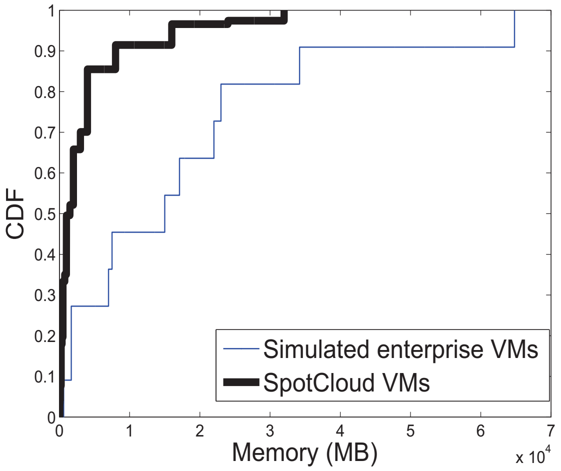

Fig. 7: Memory size

generated among the Spotcloud VMs. Fortunately, such a problem is already widely discussed in the cloud computing systems [23]. The existing data and VM placement approaches can be applied to minimize the inter-VM communications in our Spotcloud system.

Service Availability: As we have studied in [24], service availability is also a fundamental issue in our Spotcloud system. Note that Spotcloud is different from peer-to-peer networks where the customers can leave the system freely. When a seller provides resources to SpotCloud, s/he needs to claim the periods that the local resources are available. Similar to the EC2 system, customers' lease costs will be refunded when the seller failed to provide the service availability s/he previously claimed. Besides this design, we believe that the service availability should also be considered in the future pricing models.

VI. Conclusions

This paper investigated the business model of customer-provided resources sharing for cloud computing. We closely examined the user incentive and developed efficient pricing protocol for a real-world system, Spotcloud. The gaming model as well as the trace analysis validated this system as a complement of great potentials to datacenter-based cloud.

Acknowledgement

H. Wang's work is supported by a Start-up Grant from the University of Minnesota at Duluth. J. Liu's research is supported by a Canada NSERC Discovery Grant, an NSERC Strategic Project Grant, and a China NSFC Major Program of International Cooperation Grant (61120106008). K. Xu's work is supported by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry the Open Research Fund from the Key Laboratory for Computer Network and Information Integration (Southeast University, Ministry of Education, China), the Fundamental Research Funds for the Central Universities, National Key Technology R&D Program (2011BAK02B02-01), High-Tech 863 Program (2012AA111902).

References

[1] Amazon Web Service. [Online]. Available: http://aws.amazon.com/

[2] Rackspace Cloud. [Online]. Available: http://www.rackspacecloud.com/

[3] J. L. H. Wang, F. Wang and J. Groen, "Measurement and Utilization of Customer-Provided Resources for Cloud Computing," *Proc. IEEE INFOCOM, 2012.*

[4] M. Armbrust, R. G. A. Fox, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," University of California, Berkeley, Tech. Rep., 2009.

[5] K. Sripanidkulchai, S. Sahu, Y. Ruan, A. Shaikh, and C. Dorai, "Are Clouds Ready for Large Distributed Applications?" in *Proc. SOSP LADIS Workshop, 2009.*

[6] M. Hajjat, X. Sun, Y. E. Sung, D. Maltz, S. Rao, K. Sripanidkulchai, and M. Tawarmalani, "Cloudward Bound: Planning for Beneficial Migration of Enterprise Applications to the Cloud," in *Proc. ACM SIGCOMM, 2010.*

[7] SETI@HOME. [Online]. Available: http://setiathome.berkeley.edu/

[8] S. Garfinkel, "An Evaluation of Amazon s Grid Computing Services : EC2 , S3 and SQS," *Harvard University Tech, Rep., 2008.*

[9] E. Walker, "Benchmarking amazon EC2 for high-performance scientific computing," *Proc. USENIX Login, 2008.*

[10] A. Li and X. Yang, "CloudCmp: Comparing Public Cloud Providers," *Proc. ACM/USENIX IMC, 2010.*

[11] J. S. Ward, "A Performance Comparison of Clouds: Amazon EC2 and Ubuntu Enterprise Cloud," *Proc. SICSA DemoFEST, 2009.*

[12] T. Wood, P. Shenoy, and Arun, "Black-box and Gray-box Strategies for Virtual Machine Migration," in *Proc. USENIX NSDI, 2007.*

[13] V. Shrivastava, P. Zerfos, K. Lee, H. Jamjoom, Y. Liu, and S. Banerjee, "Application-aware Virtual Machine Migration in Data Centers," in *Proc. IEEE INFOCOM, 2011.*

[14] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes," *IEEE Communications Surveys and Tutorials, 7(2):72-93 , 2005.*

[15] Seller API and Third Party Provider Integration Guide. [Online]. Available: http://spotcloud.com/fileadmin/docs/SpotCloudProviderGuide.pdf

[16] Amazon EC2 Spot Instances. [Online]. Available: http://aws.amazon.com/ec2/spot-instances/

[17] K. Zhu, "Information Transparency of Business-to-Business Electronic Markets: A game-Theoretic Analysis," *Management Science, 50(5):670-685 , 2004.*

[18] M. Beckmann, "Bertrand-Edgeworth Duopoly Revisited," *In: Henn R (ed) Operations Research Verfahren III, Meisenheim: Sonderdruck, Verlag Anton Hain, 55-68, 1965.*

[19] J. Farrell and E. Maskin, "Renegotiation in Repeated Games," *Journal of Economic Theory, 1(4):327-360 , 1989.*

[20] M. Kandori, "Introduction to Repeated Games with Private Monitoring," *Journal of Economic Theory, 102(1):1?5, 2002.*

[21] D. Abreu and A. Rubinstein, "The Structure of Nash Equilibrium in Repeated Games with Finite Automata," *Journal of the Econometric Society, 56(6):1259-1281 , 1988.*

[22] N. Papatheodoulou and N. Sklavos, "Architecture & System Design Of Authentication, Authorization, & Accounting Services," in *Proc. IEEE EUROCON, 2009.*

[23] B. Palanisamy, A. Singh, L. Liu, and B. Jain, "Purlieus: Locality-aware Resource Allocation for MapReduce in A Cloud," in *International Conference for High Performance Computing, Networking, Storage and Analysis*, 2011.

[24] H. Wang, F. Wang, J. Liu, D. Wu, and Q. Lin, "Resource Provisioning on Customer-Provided Clouds: Optimization of Service Availability," in *Proc. IEEE ICC, 2013.*