Measurement and Utilization of Customer-Provided Resources for Cloud Computing

Haiyang Wang
Simon Fraser University
British Columbia, Canada
Email: hwa17@cs.sfu.ca

Feng Wang
Simon Fraser University
British Columbia, Canada
Email: fwa1@cs.sfu.ca

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

Justin Groen
Enomaly Inc.
Ontario, Canada
Email: justin@enomaly.com

***Abstract*—Recent years have witnessed *cloud computing* as an efficient means for providing resources as a form of utility. Driven by the strong demands, such industrial leaders as Amazon, Google, and Microsoft have all offered practical cloud platforms, mostly datacenter-based. These platforms are known to be powerful and cost-effective. Yet, as the cloud customers are pure consumers, their local resources, though abundant, have been largely ignored.**

In this paper, we for the first time investigate a novel customer-provided cloud platform, *SpotCloud*, through extensive measurements. Complementing data centers, SpotCloud enables customers to contribute/sell their private resources to collectively offer cloud services. We find that, although the capacity as well as the availability of this platform is not yet comparable to enterprise datacenters, SpotCloud can provide very flexible services to customers in terms of both performance and pricing. It is friendly to the customers who often seek to run short-term and customized tasks at minimum costs. However, different from the standardized enterprise instances, SpotCloud instances are highly diverse, which greatly increase the difficulty of instance selection. To solve this problem, we propose an instance recommendation mechanism for cloud service providers to recommend short-listed instances to the customers. Our model analysis and the real-world experiments show that it can help the customers to find the best trade off between benefit and cost.

I. Introduction

Cloud computing has recently attracted a substantial amount of attentions from both industry and academia [1][2][3][4][5]. The emergence of cloud computing as an efficient means of providing computing as a form of utility can already be felt with the burgeoning of cloud service companies. The existing cloud platforms are known to be powerful and cost-efficient to run many services. However, as the cloud customers are pure consumers, their local resources have been largely ignored especially considering the fast growing of personal computing capacities.

In this paper, we take a first step towards the potential of a customer-provided platform for cloud computing: Enomaly's SpotCloud [6]. This newborn platform has already attracted an increasing number of users world wide. Different from most of the existing *enterprise clouds* (provided by enterprise datacenters), SpotCloud allows customers to contribute/sell their own idle computing resources and build their own cloud platforms. In this way, the customers are no longer pure consumers but can also gain benefits while interacting with cloud services. Unlike grid computing [7], this platform is not deployed by service providers. Instead, it is simply formed/organized by self-motivated customer resources. The customers can earn profit when their instances are used by others. This distinguishes the customer-provided clouds from peer-to-peer networks which have no clear economic model.

However, these features also raise many new challenges such as the availability, security and instance selection. More importantly, it is not clear that what kind of applications are suitable for this platform and how cloud customers should select the instances to serve their applications. To answer these questions, we closely examine this customer-provided cloud platform in its computing capacity, server availability, network performance and the pricing model. We find that it is not yet ready to serve long-term applications (for example, the web service) or some CPU sensitive tasks. However, it provides very flexible choices (in terms of both performance and pricing) and is very friendly to individual customers when they seek to run short-term tasks at minimum costs.

The customer-provided instances are highly diverse in terms of their performance and cost. When the customers need to lease multiple cloud instances, their performance/cost highly depends on their instance selection. Unfortunately, most buyers do not have enough knowledge to select the right instances or even decide how many instances they need to use. Therefore, we develop an instance recommendation model to facilitate cloud providers to recommend a set of short-listed instances to the customers. To understand its effectiveness, we applied SeedBox service [8] in our real-world experiment. Such a content delivery service is not the classic service that well matches enterprise cloud platforms. Yet, it potentially works well on the SpotCloud platform given the short service duration and the distributed resource demands. The experimental results show that our solution can help the customers find the best trade off between benefit and cost; in particular, the customers can get better performance while minimizing their cost at the same time. It is also worth noting that SpotCloud is compatible with the existing enterprise clouds and it is reasonable to believe that it will become an important complement to enterprise clouds with hybrid resources.

The rest of this paper is organized as follows: In Section II, we present the related works. Based on the backgrounds that provided by Section III, we examine the real-world performance as well as the cost of SpotCloud in Section

978-1-4673-0775-8/12/$31.00 ©2012 IEEE

IV. We develop the model of instance recommendation in Section V. Section VI evaluates the benefit of this approach and compares the performance/cost of enterprise and customer-provided clouds. Some practical issues are further discussed in Section VII and Section VIII concludes the paper.

II. Related Works

The salient features of cloud computing have enticed a number of companies to enter this market [1][9][2][3][10] and have also attracted significant attention from academia [4][5][11]. There have been a series of works measuring the performance of public cloud services from diverse aspects, including computation, scaling, storage, and networking services [12] [13]. Ward et al. [14] have also compared the performance of a public cloud with that of a private cloud. A recent work from Li et al. [15] further examined the inter-datacenter network transfer through fine-grained measurement. Our work differ from them in that we focus on the measurement of resources contributed from cloud customers and the potentials of utilizing such resources, as a complement to data-center-based cloud.

Many studies have also addressed application designs that leverage cloud platforms [16] [17], or application migrations to the cloud. For example, Wu et al. [18] explored the use of cloud for Video-on-Demand applications; Kannan et al. [19] examined the optimization of home clouds for mobile devices. We have also identified potential applications that best explore the customer contributed resources.

III. SpotCloud: Backgrounds and Frameworks

SpotCloud is built on the Google App Engine [2] and the Enomaly ECP platform [20]. It provides a structured cloud capacity marketplace where service providers can sell their computing capacity to a wide range of buyers and re-sellers. Different from most of the existing *enterprise clouds* (the cloud services that are provided by enterprise datacenters), SpotCloud allows the customers to contribute/sell their own idle computing resources and build their own cloud platforms. As shown in Figure 2, we can see that SpotCloud is integrating the cloud resources from both enterprise and customer sellers. There are two kinds of users in this system: *Sellers* who sell their idle cloud computing resources and *buyers* who consume/buy these capacities. The instance sellers can dynamically define hardware profiles, location information, duration of available capacity and associated resource costs. Note that SpotCloud also provides a pricing guide to help sellers estimate the revenue they can obtain from SpotCloud. This pricing guide is based on such metrics as the capacity and the availability of the instances. An instance buyer, on the other hand, needs to create a *VM appliance*[1] using the Enomaly SpotCloud package builder or using the default appliance provided by SpotCloud [21]. After that, the buyer can upload its VM appliance using the SpotCloud management interface, and the VMs will be automatically delivered to sellers' cloud

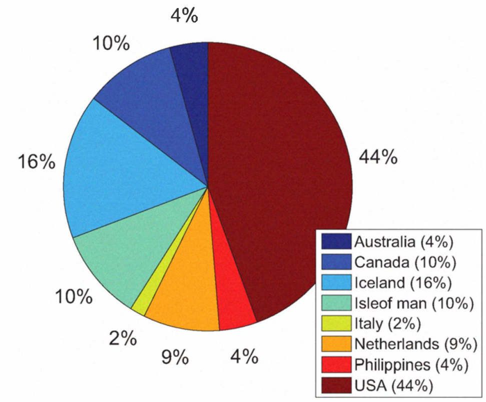


Fig. 1: Locations of SpotCloud Instances

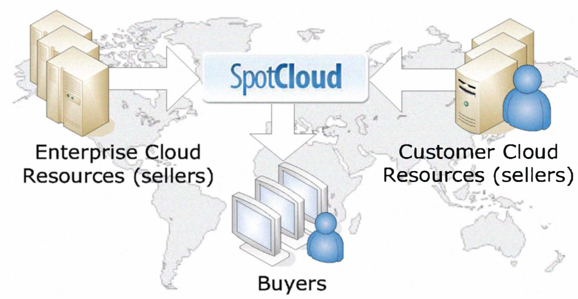


Fig. 2: Basic Framework of SpotCloud

infrastructures where the VM packages are processed according to the buyers' requirements. The SpotCloud monitors will then debit buyers on an hourly utility basis with a notification sent when credits drop below minimum threshold. Finally, the sellers will be paid directly for any capacity utilized via the SpotCloud marketplace.

It is easy to see that this platform is different from the existing enterprise clouds that consist of dedicated datacenters. In particular, the customers' local resources play an important role in SpotCloud which generally control the performance/pricing of the instances. Based on this new feature, the existing studies on enterprise clouds cannot be directly borrowed to understand such a new platform. It is not clear that what kind of applications are best suitable for this platform and how buyers should select the instances to serve their applications. To answer these questions, we investigate the customer-provided resources on SpotCloud through extensive measurements[2]. This newborn platform has already attracted attentions over the Internet; however, its scale is not yet comparable to existing enterprise cloud services. Therefore, we investigate 116 instances as a first step to understand the basic features of this platform. The physical locations of these instances are shown in Figure 1.

[1]VM appliance is a virtual machine image designed to run on a virtualization platform (e.g., VirtualBox, Xen and VMware).

[2]Our data is obtained from the management servers that deployed in the SpotCloud platform.

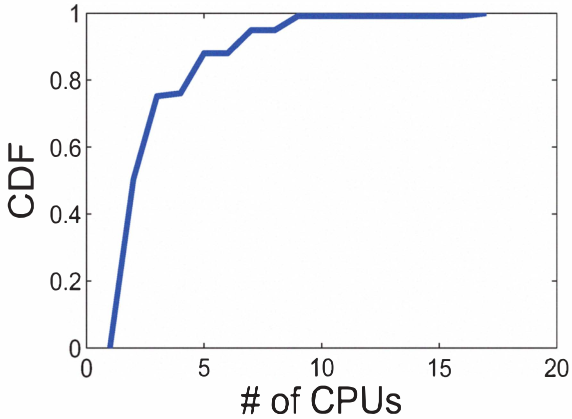


Fig. 3: # of CPUs in the instances

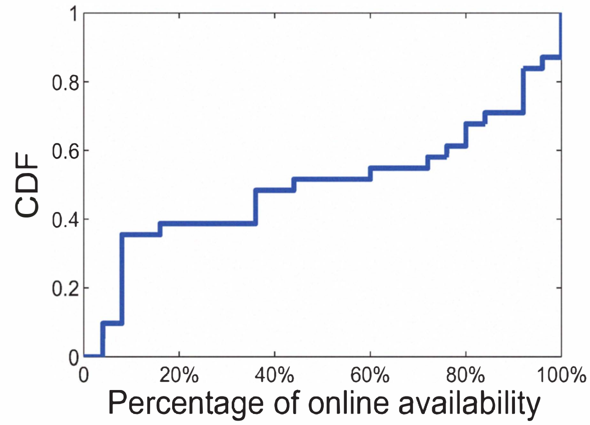


Fig. 5: Online availability of SpotCloud instances

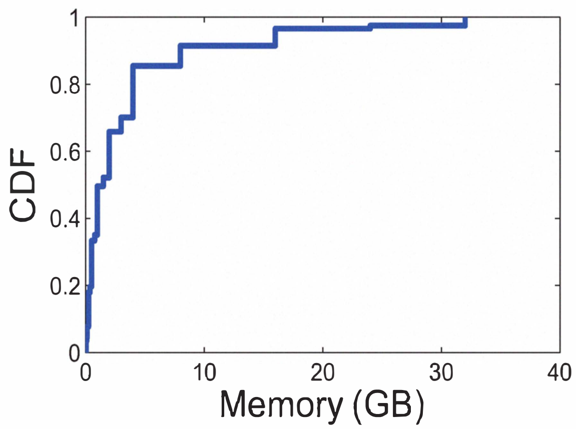


Fig. 4: Memory in the instances

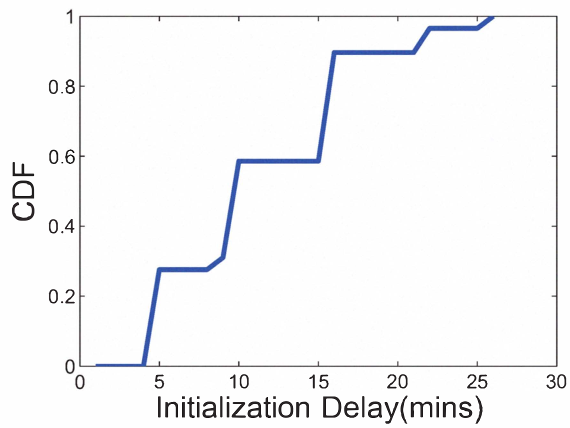


Fig. 6: Instance initialization delay

IV. Performance and Cost Measurement

We first check the number of CPUs in the SpotCloud instances. As shown in Figure 3, it is easy to see that most SpotCloud instances ($> 75\%$) possess less than 4 virtual cores. 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 instances; for example, an instance 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 of the instances in Figure 4. We can see that most (80%) instances in SpotCloud have a memory less than 5GB, which is not extra huge but is suitable to run most of the real-world tasks.

Different from enterprise servers that are known to have very high availability, the service availability in SpotCloud mostly depends on the instance sellers. Figure 5 shows the online availability of the instances for one month; 40% instances have an online availability below 20%, that is, less than 6 days in the 30-day measurement period. This availability is acceptable for short-term tasks lasting for a few hours or days. For longer tasks, SpotCloud needs to carefully assist its customers to choose proper instances.

It is also worth noting that, before a buyer can really use a cloud instance, there is a delay due to a necessary initialization process. For example, the AWS Management Console [22] shows that it generally needs 15 to 30 minutes to initialize a Windows instance on Amazon EC2 before a buyer can really connect to it. For SpotCloud, as shown in Figure 6, we can see that most instances (more than 60%) can be initialized within 10 minutes, and the maximum initialization delay is less than 27 minutes. This is considerably lower than that of Amazon EC2. The reason is that the system/user profiles of SpotCloud instances are already included in buyers' VM appliances. Note that the instances' operation systems can also be personalized by the buyers in SpotCloud. If a buyer dose not want to decide the OS type, Linux (Ubuntu 10.10) is set as a default, which indeed has even lower initialization delay than the average. We further investigate the instance throughput. As shown in Figure 7, we can see that the throughput of over 50% instances are more than 10 MB, which is good enough to deliver customers' contents to the cloud servers in normal cases.

One important feature of cloud services is that the cus-

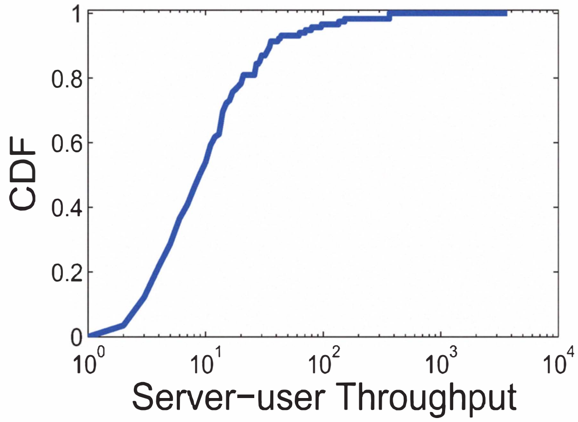


Fig. 7: Throughput between cloud server and user

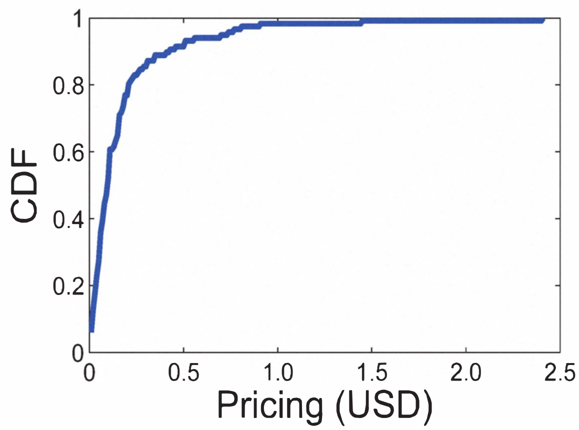


Fig. 8: Pricing of SpotCloud instances

tomers pay only for what they have used. In most of the enterprise clouds, this cost consists of two major parts: the cost of using instances, and the cost of data transfer. Their pricing model is computed/decided carefully by the enterprise service providers. However, the price of SpotCloud instances is customized by individual sellers who provide/sell their cloud capacities. As shown in Figure 8, we can see that the SpotCloud instances are mostly very cheap. Moreover, this curve is also quite smooth, indicating that the buyers have diverse options to select instances in this customer-provided cloud platform.

V. Instances Recommendation: Performance Optimization across Cloud Instances

Based on our measurement, it is easy to see that the main advantage of SpotCloud comes from its flexible and customized performance/cost that are enabled by diverse customers. However, this also brings new challenges to the cloud service, especially when the customers need to apply for multiple cloud instances at a time. Different from enterprise clouds where the instances are mostly standardized into a few classes, the instances in SpotCloud are highly diverse; for example, we can hardly even find two identical instances in the SpotCloud marketplace. In this case, when the buyers need to rent multiple cloud instances, the performance/cost will highly depend on their instance selection. Unfortunately, most buyers do not have enough knowledge to select the right instances or even decide how many instances they need to use. A good instance recommendation mechanism is therefore critical to the design of SpotCloud.

A. Modeling of Instance Recommendation

To investigate instance recommendation in SpotCloud, we apply the popular *SeedBox* service as a case study [8]. A seedbox is a private dedicated server for uploading and downloading files, where a peer-to-peer protocol like BitTorrent is used for data exchange. The file will be downloaded from BT swarms to seedbox servers via the BitTorrent protocol and finally be sent to the users via FTP. Such a content delivery service is not the classic service that well matches enterprise cloud platforms. Yet it potentially works well on the SpotCloud platform given the short service duration and the distributed resource demands.

For the instance recommendation in this case, we assume that the cloud providers can obtain some pre-knowledge of buyers' budget and task. For example, the cloud providers can obtain the torrent that the buyers want to download and the total amount of money that the buyers want to spend. Based on this information, the cloud providers will help the buyers to find a set of instances to maximize their content downloading performance and make sure that the total cost will not exceed the buyers' budget.

We use C to denote the set of cloud instances; each instance $c \in C$ has an uploading capacity u_c and a downloading capacity d_c. The price of buying an instance is denoted by w_{ins} (USD per hour), and the traffic price is denoted by w_{tra} (USD per GB). For the BitTorrent swarm[3], we use T to refer the set of peers. S refers to the set of *seeders* (the peers who have a complete copy of the file and still stay in the system to help other peers) and L refers to the set of *leechers* (peers who do not have a complete copy of the file; they are generally sharing what they have and downloading what they need) where $T = S \bigcup L$. Each peer $t \in T$ has an uploading capacity u_t and a downloading capacity d_t. This swarm serves a given file of size F.

Giving this setting, a buyer p^* is trying to buy a set of cloud instances to facilitate its downloading. The uploading and downloading capacity of this buyer is denoted by u_{p^*} and d_{p^*} respectively. We assume the bottleneck is at the edge of networks, and we use b to refer the end-to-end bandwidth.

1) Pure Seeder Case: We first discuss the simplest case when the BitTorrent swarm consists of only one seeder. We focus on improving the downloading completion time as the performance gain. Therefore, the gain of buying one cloud instance (for example, the instance $c \in C$) is given by

[3]Note that the BT protocol is supported in Amazon S3 to accelerate the content delivery. This protocol also helps most seedbox servers to obtain Internet contents.

$$Gain_1(s,c) = \frac{F}{b_{s,p^*}} - \frac{F}{min(b_{s,c} \cdot P_1, b_{c,p^*})} \quad (1)$$

$$s.t. \quad min(b_{s,c}, b_{c,p^*}) \leq min(u_s, d_{p^*}) \quad (2)$$

where P_1 is the probability of optimistic unchoking ($P_1 = \frac{1}{|C|+|T|}$). In the pure seeder case, P_1 is equal to 1. The first part of $Eq.1$ refers to the downloading time without cloud assistance and the second part refers to the downloading time with the help of cloud instance c. Note that $Gain_1(s,c) \leq 0$ means that we cannot benefit from using cloud instance c. Based on this equation, we can further get the performance gain of renting multiple (for example $|C^1|$) cloud instances as follows (where $C^1 \subseteq C$):

$$Gain_1(s,C^1) = \frac{F}{b_{s,p^*}} - \frac{F}{\sum_{c \in C^1}\{min(b_{s,c} \cdot P_1, b_{c,p^*})\}} \quad (3)$$

$$s.t. \quad min(b_{s,c}, b_{c,p^*}) \leq min(u_s, d_{p^*}) \quad (4)$$

The second part of $Eq.3$ refers to the downloading time of obtaining content F with the help of $|C^1|$ cloud instances. Let $Time(F,C^1) = \frac{F}{\sum_{c \in C^1}\{min(b_{s,c}, b_{c,p^*})\}}$, the total cost of renting $|C^1|$ cloud instances is given by

$$Cost_1(s,C^1) = Time(F,C^1) \sum_{c \in C^1} w_{ins} + F \cdot w_{tra} \quad (5)$$

where the first part of $Eq.5$ refers to the instance cost and the remaining part refers to the traffic cost. Note that in the pure seeder case, the cloud instances only need to upload the contents to the peer p^*. Therefore, the traffic cost is equal to $F \cdot w_{tra}$.

2) Pure Leecher Case: We now discuss another case when the BitTorrent swarm consists of only leechers. The analysis of these two cases will help us obtain the general model of the instance recommendation problem.

Assuming that there is only one leecher l in the BitTorrent swarm. A buyer p^* wants to rent $|C^2|$ cloud instances to assist its downloading. In this case, the performance gain of renting a cloud instance c is given by:

$$Gain_2(l,c) = \frac{F}{b_{l,p^*}} - \frac{F}{min(r_{l,c}, b_{c,p^*})} \quad (6)$$

where $r_{l,c}$ refers to the expected downloading rate between BitTorrent swarm and the cloud instances:

$$r_{l,c} = b_{l,c} \cdot P_1 + b_{l,c} \cdot P_2 \quad (7)$$

where P_1 is the probability of optimistic unchoking ($P_1 = |C|/|T|$) and P_2 is the probability of regular unchoking, and $P_2 \approx \left(\frac{LogN}{N}\right)^{|T|}$ [23], where N is the number of pieces of the served file F and $|T|$ is the total number of peers. Therefore, the performance gain of renting $|C^2|$ cloud instances is given by

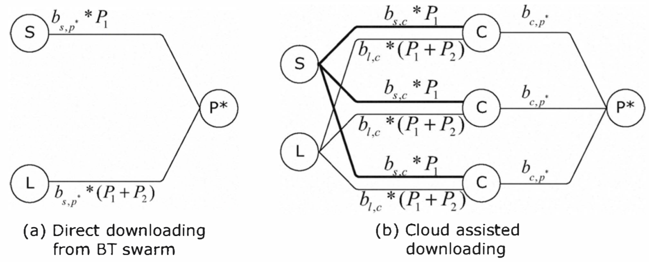


Fig. 9: Example of performance gain

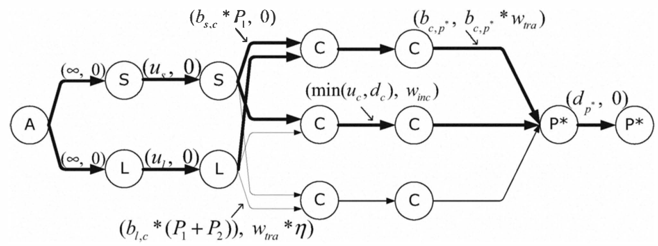


Fig. 10: Flow network transformed from Figure 9(b)

$$Gain_2(l,C^2) = \frac{F}{b_{l,p}} - \sum_{c \in C^2} \frac{F}{min(r_{l,c}, b_{c,p^*})} \quad (8)$$

$$s.t. \quad min(r_{l,c}, b_{c,p}) \leq min(u_l, d_{p^*}) \quad (9)$$

We use η to refer the trade off between uploading and downloading in the swarm, where $\eta = 1-(M_{down}/M_{up}+M_{down})$; M_{down} is the downloading amount and M_{up} is the uploading amount. $\eta \in [0,1)$ where $\eta = 0$ refers the free ridding case. Let $Time(F,C^2) = \sum_{c \in C^2} \frac{F}{min(r_{l,c}, b_{c,p^*})}$, the total cost of buying $|C^2|$ cloud instances is given by

$$Cost_2(l,n_2) = Time(F,C^2) \sum_{c \in C^2} w_{ins} + F \cdot w_{tra} + F \cdot w_{tra} \cdot \eta \quad (10)$$

Note that in the customer-provided clouds where the traffic is not charged, w_{tra} will be set to 0.

3) General Case With Multiple Seeders/Leechers: Based on the analysis of above two cases, we can merge them together and obtain the model for instance recommendation across all the peers in both seeder set S and leecher set L aiming to maximize the total $Benefit$ and to minimize the total $Cost$:

$$Benefit = \left\{\sum_{s \in S} Gain_1(s,C^1) + \sum_{l \in L} Gain_1(l,C^2)\right\} \quad (11)$$

$$Cost = \left\{\sum_{s \in S} Cost_1(s,C^1) + \sum_{l \in L} Cost_2(l,C^2)\right\} \quad (12)$$

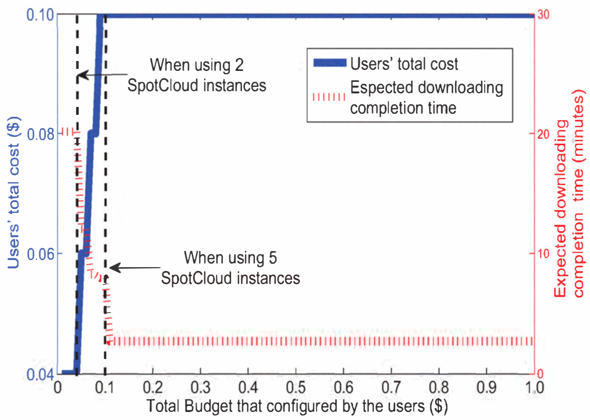


Fig. 11: Simulation result of SpotCloud instance recommendation

$$
\begin{aligned}
s.t. \quad min(b_{s,c}, b_{c,p^*}) &\leq min(u_s, d_{p^*}) & (13)\\
min(r_{l,c}, b_{c,p}) &\leq min(u_l, d_{p^*}) & (14)\\
\sum_{c\ inC} b_{c,p^*} &\leq d_{p^*} & (15)\\
Benifit > 0 \quad and &\quad Cost \leq Q & (16)
\end{aligned}
$$

Besides the existing constraints in $Gain_1$ and $Gain_2$ (Eq.13 and Eq.14), $Eq.15$ and $Eq.16$ show the extra constraints after the merging; where $Eq.15$ is the bandwidth constraint and Q is the buyer's total budget and this total cost cannot exceed Q; $Benefit > 0$ means that the use of cloud instances should at least accelerate the downloading of peer p^*.

B. Problem Transformation and Analysis

Based on this model, it is easy to see that the cloud instances are working together like an amplifier between peer-to-peer swarms and buyer p^*. Figure 9 gives an illustrative example of the performance gain. P_1 and P_2 refers to the probability of optimistic unchoking and regular unchoking respectively. If we only compare the end-to-end downloading rate between seeder S and buyer P^*, we can see that the buyer can achive better downloading rate unless $b_{s,p^*} \geqq 3b_{s,c}$. Considering the fact that the cloud instances are mostly high performance servers, the case of $b_{s,p^*} \geqq 3b_{s,c}$ can hardly happen in real-world[4].

To solve the instance recommendation problem, we convert it into a minimum cost maximum flow problem in a flow network. The nodes' bandwidth constraints are transferred into edge capacities, such as $(u_s, 0)$ on seeder s. Without loss of generality, we also give edges directions and add a virtual node A as the source of the flow network. Figure 10 shows a conversion for Figure 9(b)where the darker lines show an example of the solution. Each edge has a capacity and a cost marked as $(capacity, cost)$.

The objective is to maximize the flow (end-to-end downloading rate) and minimize the flow cost, which can be addressed by the classical Ford-Fulkerson algorithm. Note that the main inputs of this model are two matrixes: edge capacity matrix and edge cost matrix. Different cloud providers can use their own server capacities and cost models to generate these two matrices. For example in SpotCloud, $w_{tra} = 0$ and the cost of most edges will equal to zero. For a given file of size F, this is an equivalent problem with Eq.11-16 [5]. A solution to this problem gives the buyers clear guide on the instance selection. For example, in Figure 10 (where the paths of maximal flows are marked in dark lines), buyer P^* should select 2 cloud instances to assist its downloading.

Figure 11 shows a simulated result for SpotCloud. In this simulation, we use our measurement trace and the pricing model to generate the edge capacity matrix and edge cost matrix. η is set to be 0.2 and the file size F is 1 GB. In Figure 11, the solid line refers to buyers' total cost(cannot exceed the buyers' budget) and the dotted line refers to the expected downloading completion time. We can see that in SpotCloud, when the buyer's budget is set to be less than 0.05 USD, our algorithm shows that the buyers can use 2 cloud instances to achieve the total downloading time of 20 minutes. However, when the buyer is willing to pay more, say 0.1 USD, it can then use 5 instances at a time and the total downloading time will be sharply decreased to 2.8 minutes. We can also see that the downloading completion time cannot be further decreased even when the buyers want to pay more. This minimum downloading completion time is bounded by the maximum flow (maximum downloading rate) in the flow network and using 5 instances is the optimal solution in this case. Based on these results, SpotCloud can recommend a list of instances with 2 options for the buyers: (1) spend 0.03 USD and 20 minutes to rent two instances; or (2) spend 0.1 USD and less than 3 minutes to rent five instances.

VI. Experiment and Evaluation

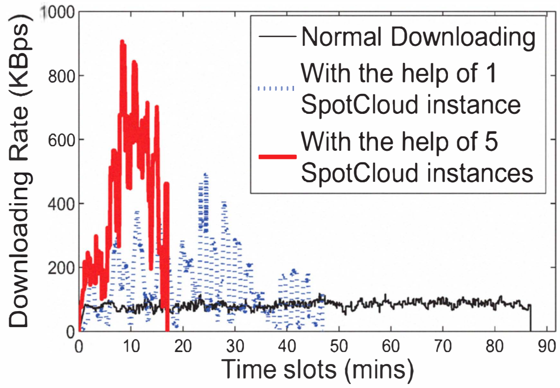
In this section, we will further discuss the possible gain of the instance recommendation model via real-world experiments. In particular, we apply the SeedBox service on both EC2 and SpotCloud platforms to understand whether SpotCloud has the potential to complement the enterprise clouds. The configuration of our experiment is as follows:

We deployed 5 nodes in Planet Lab to simulate an unpopular BT swarm[6]. This swarm consists of 1 seeder and 4 leechers; these peers are serving a content of size $384MB$ (this content size is limited by the storage constraints of the Planet Lab servers). We use a normal PC in our campus to simulate buyer P^*. We select a sample set of 10 instances from Amazon EC2 including 4 micro instances with hourly price of 0.02 USD, $613M$ memory and 2 EC2 compute units; 3 small instances with hourly price of 0.085 USD, $1.7GB$ memory and 1 EC2 compute unit; 3 large instances with hourly price of 0.34 USD, $7.5G$ memory and 4 EC2 compute units. We have tried

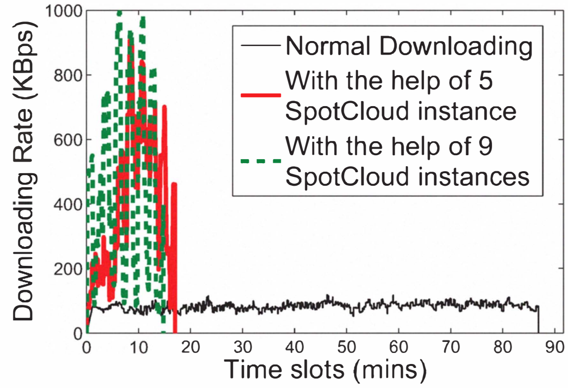
[4]Without loss of generality, we assume that $b_{c,p^*} = b_{s,c}$ in this case.

[5]The proof can be found in our technical report[24]

[6]This is based on the service feature of SeedBox. If the peer can obtain very high downloading speed from a popular swarm (with more peers), it is not necessary to apply SeedBox service to accelerate the downloading

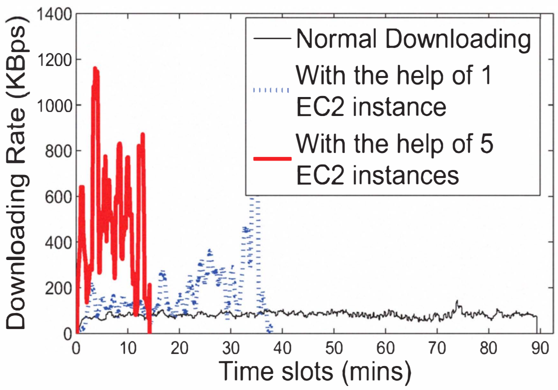


(a) Downloading performance of using 0, 1, 5 instances

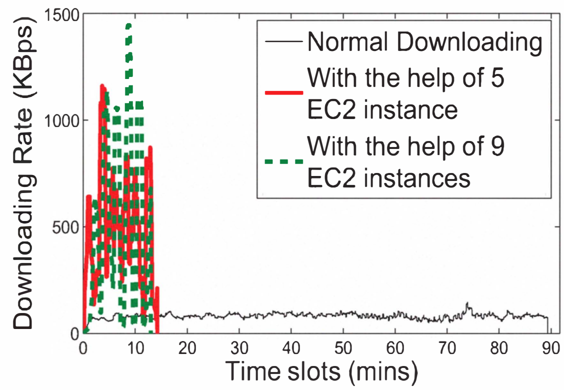


(b) Downloading performance of using 0, 5, 9 instances

Fig. 12: Real-world experiments of using SpotCloud instance



(a) Downloading performance of using 0, 1, 5 instances



(b) Downloading performance of using 0, 5, 9 instances

Fig. 13: Real-world experiments of using Amazon EC2 instance

to use different types of EC2 instances in the experiment, and we found that using more powerful instances will not affect the results of our instance recommendation. The set of SpotCloud instance also consists of 10 randomly selected instances and the instance information is shown in Table I. The buyers' total budget is set to be 0.5 USD on EC2 and 0.1 USD on SpotCloud. We also have tested these budgets in the experiment for several times; these two budgets can achieve similar downloading performance on both platforms and thus give us clear results for comparison. Based on this controlled environment, we can successfully obtain the capacity of cloud instances in both platforms. After running our instance recommendation algorithm, we have the modeling results as follows:

For the EC2 platform, the buyers have two options: (1) Use one small instance for 22 minutes, which will cost 0.097 USD; or (2) Use one small and four micro instances at the same time for 8 minutes, which will cost 0.377 USD. For SpotCloud platform, the buyers also have two options: (1) Use instance $\#4$ for 26 minutes, which will cost 0.02 USD; (2) Use instance $\#1$, $\#2$, $\#4$, $\#5$ and $\#6$ at the same time for 9 minutes, which will cost 0.047 USD.

Based on these recommendation results, we further validate the real-world performance/cost on both EC2 and SpotCloud platforms by answering following questions: First, whether the instance recommendation model can find good trade off between the benefit and the cost? Second, whether the buyers can experience better performance by using more instances beyond the recommendation advices? and Third, comparing to enterprise clouds, whether the customer-provided instances can give more benefits to the buyers?

To answer these questions, we perform experiments on both EC2 and SpotCloud platforms to test the downloading performance as well as the cost for the buyers. Figure 12 shows the downloading rate as well as the downloading completion time when the buyers are using different numbers of SpotCloud instances based on the recommendation. As shown in Figure 12a, we can see that if the buyers do not use any cloud instance to help their downloading, the downloading completion time will be 86.95 minutes, and the downloading rate is around $80KBps$. When s/he follows the first recommendation from SpotCloud (using instance #4

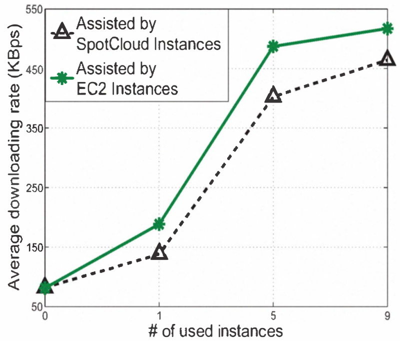


Fig. 14: Comparison of average downloading rate

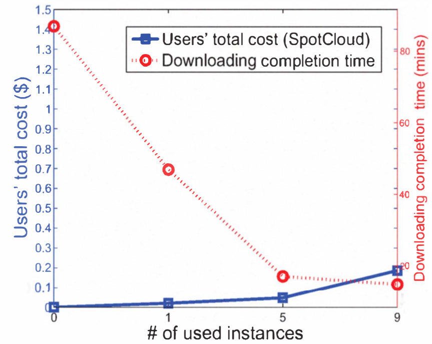


Fig. 15: Trade off between benefit and cost (SpotCloud)

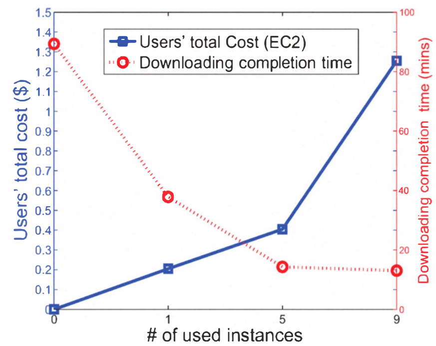


Fig. 16: Trade off between benefit and cost (Amazon EC2)

to help the downloading), the downloading completion time will be decreased to 46.9 minutes and the downloading rate will be increased to around $200KBps$. Moreover, when the buyers follow the second recommendation (using 5 instances to help the downloading). The downloading completion time will be decreased to 17.0 minutes and the downloading rate can sometimes reach to $900KBps$.

However, if the buyers do not want to follow the recommendation advices and try to use 9 instances for downloading, as shown in Figure 12b, we can see that the downloading completion time is still slightly reduced from 17.0 minutes to 14.8 minutes. If we compare this benefit with buyers' cost, as shown in Figure 15. It is easy to see that this benefit is not proportional to the buyers' cost. In particular, the buyers can spend 0.047 USD (using 5 instance) to reduce the downloading completion time by 80% (from 86.95 minutes to 17.0 minutes); yet, when the buyers further increase their cost by 0.14 (total cost becomes 0.187) USD, the downloading completion time can only be decreased by 13% (from 17 minutes to 14.8 minutes). Figure 12b also confirms that the downloading completion time can no longer be effectively reduced when the buyers applied more than 5 instances.

On the other hand, Figure 13 shows the downloading rate as well as the downloading completion time using different numbers of EC2 instances. We can see that when using a similar number of instances, the downloading completion times of using EC2 instances are slightly shorter than that of using SpotCloud. This can be further confirmed in Figure 14 which shows that SpotCloud can provide similar performance when compared to the enterprise servers. From Figure 16, we can see that the cost of using enterprise instances is generally quite high. For example, it will cost the buyers 0.205 USD to complete the downloading in 37.8 minutes, and 0.450 USD to further reduce it to 14.3 minutes. Comparing to Figure 15 (the SpotCloud case), the buyers only need to spend 10% of this cost to achieve similar downloading completion time by using SpotCloud instances. Note that the real-world costs are slightly higher than the modeling results; this minor difference is due to the value of η and will not bias our investigation.

TABLE I: Selected SpotCloud Instances

Index	Price	# of CPU	Memory	Country
1	$0.002	1	256MB	Isle of Man
2	$0.005	1	256MB	United States
3	$0.020	4	1GB	United States
4	$0.020	1	4GB	United States
5	$0.010	2	8GB	Isle of Man
6	$0.010	1	512MB	United States
7	$0.019	1	256MB	Netherland
8	$0.041	1	256MB	Iceland
9	$0.238	4	8GB	Iceland
10	$0.06	1	512MB	Poland

VII. Further Discussions

This paper takes a first step towards the measurement and the utilization of customer-provided cloud platform: SpotCloud. There are still many open issues that can be further explored.

Service Availability: To enable enterprise-level services in SpotCloud, we have to ensure high service availability when integrating customers' resources. Different from datacenters, there is no guarantee that a particular customer's local resources will be always online for cloud computing. Through trace-analysis and algorithm design, we are now working on a smart resource provisioning across dynamic customer-provided resources. Our initial results suggest that highly stable service availability that is comparable with state-of-the-art datacenters is possible with the distributed and dynamic resources.

Customer Incentive: Another critical challenge is to offer incentive for a customer to contribute her/his resources or to utilize others'. The problem is further complicated given that the customers are highly heterogeneous, making a coarse-grained pricing model used by the existing cloud providers hardly work. Although the SpotCloud has already attracted many customers, it remains unclear whether some customers are selling the instances with the prices that lower than their running costs. Therefore, the design of a better incentive/business model is still necessary for our systems and remains an open issue.

Hybrid cloud platform with both enterprise and customer-provided resources: It is worth emphasizing that SpotCloud serves as a complement to data-center-based enterprise cloud services. While it is flexible and inexpensive for certain services, we do not expect that it can well serve all types of services with comparable performance and quality as enterprise cloud. It would be interesting to form a hybrid cloud platform that leverages both enterprise and customer-provided resources to achieve the best flexibility, scalability, and cost-effectiveness. However, the instance selection strategy and the pricing model have to be substantially revised to best coordinate the two distinct types of resources, not to mention the service availability and security concerns.

VIII. Conclusions

In this paper, we for the first time examined the potentials as well as the challenges in a customer-provided cloud platform, SpotCloud. We found that, although the capacity as well as the availability of this platform are not yet comparable to the enterprise datacenters, SpotCloud can provide very flexible choices to customers in terms of both performance and pricing. To better utilize the costumer-provided resources, we also proposed an instance recommendation mechanism to address the instance selection in such a newborn platform. Our model analysis and the real-world experiments showed that it can help the instance buyers to find the best trade off between benefit and cost. The analysis also validated SpotCloud as a complement of great potentials to datacenter-based clouds.

Acknowledgement

This research was supported by a Canadian NSERC Discovery Grant, a Discovery Accelerator Supplements Award, an NSERC Engage Grant, a MITACS Project Grant, and a China NSFC Major Program of International Cooperation Grant (61120106008).

References

[1] Amazon Web Service. [Online]. Available: http://aws.amazon.com/
[2] Google AppEngine. [Online]. Available: http://code.google.com/appengine/
[3] Microsoft Windows Azure. [Online]. Available: http://www.microsoft.com/
[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] SpotCloud. [Online]. Available: http://www.spotcloud.com/
[7] F. Berman, G. Fox, and A. J. Hey, "Grid Computing: Making The Global Infrastructure a Reality," *John Wiley & Sons, April 8, 2003, ISBN 047085319*.
[8] SeedBox. [Online]. Available: http://www.superseedbox.com/
[9] GoGrid Cloud Hostin. [Online]. Available: http://gogrid.com/
[10] Rackspace Cloud. [Online]. Available: http://www.rackspacecloud.com/
[11] M. Hajjat, X. Sun, Y.-W. 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*.
[12] S. Garfinkel, "An Evaluation of Amazon s Grid Computing Services : EC2 , S3 and SQS," *Harvard University Tech, Rep., 2008*.
[13] E. Walker, "Benchmarking amazon EC2 for high-performance scientific computing," *Proc. USENIX Login, 2008*.
[14] J. S. Ward, "A Performance Comparison of Clouds: Amazon EC2 and Ubuntu Enterprise Cloud," *Proc. SICSA DemoFEST, 2009*.
[15] A. Li and X. Yang, "CloudCmp: Comparing Public Cloud Providers," *Proc. ACM/USENIX IMC, 2010*.
[16] Y. Seung, T. Lam, L. E. Li, and T. Woo, "Seamless Scaling of Enterprise Applications into The Cloud," in *Proc. IEEE INFOCOM, 2011*.
[17] U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh, "Kingfisher: Cost-aware Elasticity in the Cloud," in *Proc. IEEE ICDCS, 2011*.
[18] Y. Wu, C. Wu, B. Li, X. Qiu, and F. Lau, "CloudMedia: When Cloud On Demand Meets Video On Demand," in *Proc. IEEE ICDCS, 2011*.
[19] S. Kannan, A. Gavrilovska, and K. Schwan, "Cloud4Home – Enhancing Data Services with Home Clouds," in *Proc. IEEE ICDCS, 2011*.
[20] Enomaly ECP. [Online]. Available: http://www.enomaly.com/
[21] SpotCloud Appliance Creation Guide (For Buyers). [Online]. Available: http://www.spotcloud.com/fileadmin/docs/
[22] AWS Management Console. [Online]. Available: http://aws.amazon.com/console/
[23] D. Qiu and R. Srikant, "Modeling and Performance Analysis of Bit Torrent-Like Peer-to-Peer Networks," in *Proc. ACM SIGCOMM, 2004*.
[24] H. Wang, F. Wang, and J. Liu, "Instance Recommandation for User-assisted Clouds: Design and Optimization," *Simon Fraser University Tech, Rep., 2011*.