

# Towards Optimized Fine-Grained Pricing of IaaS Cloud Platform

<sup>1</sup>Mrs. S.JYOTHIREDDY, <sup>2</sup> Mr. G.LAKPATHI

<sup>1</sup> Pursuing M.Tech(CSE) from Jagruti Institute of Engineering and Technology

<sup>2</sup> Assistant Professor, Department of Computer Science and Engineering, Jagruti Institute of Engineering and Technology, Telangana State, India.

**Abstract:** In Cloud many pricing schemes in IaaS platform are already proposed with pay-as-you-go and subscription/spot market policy to guarantee service level agreement, it is still inevitable to suffer from wasteful payment because of coarse grained pricing scheme. In this paper, we investigate an optimized fine-grained and fair pricing scheme. Two tough issues are addressed: (1) the profits of resource providers and customers often contradict mutually; (2) VM-maintenance overhead like startup cost is often too huge to be neglected. Not only can we derive an optimal price in the acceptable price range that satisfies both customers and providers simultaneously, but we also find a best-fit billing cycle to maximize social welfare (i.e., the sum of the cost reductions for all customers and the revenue gained by the provider). We carefully evaluate the proposed optimized fine-grained pricing scheme with two large-scale real-world production traces (one from Grid Workload Archive and the other from Google data center). We compare the new scheme to classic coarse-grained hourly pricing scheme in experiments and find that customers and providers can both benefit from our new approach. The maximum social welfare can be increased up to 72:98% and 48:15% with respect to DAS-2 trace and Google trace respectively.

**Keywords:** Fine grained, IaaS, DAS, Vm(virtual machine)

## 1. INTRODUCTION

CLOUD computing is taking the computing world by storm, as indicated in a report by Forrester Research [1]: the global cloud market is expected to reach \$241 billion in 2020, compared to \$40.7 billion in 2010, a six-fold increase. Infrastructure-as-a-Service (IaaS) has become a very powerful paradigm to provision elastic compute resources. With an explosive growth of virtualization technology in recent years, more and more scientists are migrating their applications to the IaaS environment [2], [3]. Deelman et al. [4], for example, confirmed the

feasibility of running astronomy application on Amazon EC2 [2]. Marathe et al. [5] made a comparative study of running high performance computing (HPC) applications on the cluster and cloud. In general, there are two serious issues in deploying and provisioning virtual machine (VM) instances over IaaS environment, refined resource allocation [6] and precise pricing for resource renting [7]. Refined resource allocation is usually implemented by deploying VM instances and customizing their resources on demand, which impacts the performance of VMs to complete customers' workload. Precise pricing

is also known as Pay-as-you-go, which involves multiple types of resources like CPU, memory, and I/O devices [8]. Pricing is a critical component of the cloud computing because it directly affects providers' revenue and customers' budget [9]. How to design an appropriate pricing scheme which can make both providers and customer's satisfied is becoming a major concern in IaaS environment. In Amazon EC2, for example, the smallest pricing time unit of an on-demand instance is one hour [2]. Such a coarse-grained hourly pricing is likely to be economically inefficient for short-job users. For instance, users have to pay for full hour cost even their jobs just consumed the resources with a small portion (such as 15 minutes) of the one-hour period. Such a phenomenon is called partial usage waste, which appears very often as cloud jobs are quite short in general [10]. Based on the recent characterization of Cloud environment versus Grid systems [10], cloud jobs are usually much shorter (such as dozens of minutes) than Grid jobs (such as dozens of hours or days). This will induce serious partial usage waste issue. As illustrated in Fig. 1, the current hourly pricing scheme probably induce idle charged resources especially for short jobs, while the fine-grained pricing scheme not only makes users pay less but also makes provider gain more due to the optimization of unit price for the same service time and more users served. Some highly skilled users aggregate their jobs into a long job [11] or unite with other users under a brokerage service [12], [13] to utilize the whole instance hour. But it either requires professional knowledge which is probably a little bit too hard for ordinary users, or trusts other brokers whose credibility's can hardly be guaranteed.

## 2. SYSTEM STUDY

### 2.1. Existing System:

In general, there are two serious issues in deploying and provisioning virtual machine (VM) instances over IaaS environment, refined resource allocation and precise pricing for resource renting. Refined resource allocation is usually implemented by deploying VM instances and customizing their resources on demand, which impacts the performance of VMs to complete customers' workload. Precise pricing is also known as Pay-as-you-go, which involves multiple Types of resources like CPU, memory, and I/O devices. Pricing is a critical component of the cloud computing because it directly affects providers' revenue and customers' budget.

#### Disadvantages of Existing System:

- In the pay-as-you-go cloud pricing, short-job users have to pay more than what they actual use.
- Data usage wastage more.
- Cloud providers cannot give the services to more users.

### 2.2. Proposed System:

An appropriate pricing scheme which can make both providers and customers satisfied is becoming a major concern in IaaS environment. In Amazon EC2, for example, the smallest pricing time unit of an on-demand instance is one hour . Such a coarse-grained hourly pricing is likely to be economically inefficient for short-job users. For instance, users have to pay for full hour cost even their jobs just consumed the resources with a small portion (such as 15 minutes) of the one-hour period. Such a phenomenon is called partial usage waste, which appears very often as cloud jobs are quite short in general. Based on the recent characterization of Cloud environment versus Grid systems, cloud jobs are usually much shorter (such as dozens of minutes) than Grid jobs (such as dozens of hours or days). This will

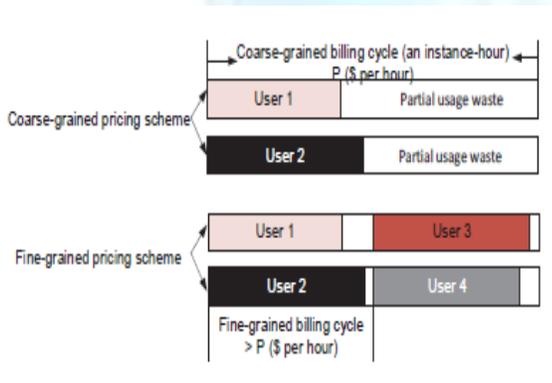
induce serious partial usage waste issue. The current hourly pricing scheme probably induce idle charged resources especially for short jobs, while the fine-grained pricing scheme not only makes users pay less but also makes provider gain more due to the optimization of unit price for the same service time and more users served.

**Advantages of proposed system:**

- In proposed we reduced data usage waste.
- Short-job users are satisfied in payment cost.
- Optimal price point can satisfy both users and providers with maximized total utility.

**3. SYSTEM ARCHITETURE**

**Architecture Diagram:**



**System Implementation**

1. Resource pricing scheme
2. Fine Grained Model
3. Coarse Grained Model
4. Cloud Time Allocating

**Resource pricing scheme:**

The existing classic IaaS cloud pricing schemes, and then analyze the partial usage waste issue, and finally formulate our optimized fine-grained pricing model by taking VM maintenance overhead into consideration. Our main objective is to satisfy both customers and providers, and

reach maximum social welfare meanwhile. In recent times, the pricing schemes broadly adopted in IaaS cloud market can be categorized into three types: pay-as-you-go offer, subscription option and spot market. Under the pay as- you-go scheme, users pay a fixed rate for cloud resource consumption per billing cycle (e.g., an hour) with no commitment. On-Demand Instances are often used to run short-jobs or handle periodic traffic spikes. In the subscription scheme, users need to pay an upfront fee to reserve resources for a certain period of time (e.g., a month) and in turn receive a significant price discount. The billing cycles in the subscription scheme are relatively long compared to the pay-as you- go scheme, and can be one day, one month, or even several years . Therefore, it is suitable for long-running jobs (like scientific computing.

**Fine Grained Model:**

We propose a novel optimized fine grained pricing scheme to solve the above issues. The objective of our work is two-fold, with regard to the classic coarse-grained pricing scheme and inevitable VM-maintenance overhead. On one hand, we aim to derive an acceptable pricing range for both customers and providers, and also derive an optimal price that satisfies both sides with maximized total utility. On the other hand, we hope to find a best-fit billing cycle to maximize the social welfare related to both sides. There are three key terms in our fine-grained pricing scheme: resource bundle, time granularity and unit price. The resource bundle serves as a kind of container to execute task workloads based on user demands. The time granularity is defined as the minimum length in pricing the rented resources. The unit price specifies how much the user needs to pay per

time granularity for the resource consumption. We give an example to illustrate the above terms. In Amazon EC2's pay-as-a small on-demand VM instance (Unix/Linux usage). In this example, the VM instance which bundles CPU, RAM, data storage and bandwidth together is the resource bundle. The time granularity is one hour and the unit price is equal to \$0:044. As another example, in Cloud Sigma, the resource bundle is not an instance but just some type of resource like CPU or RAM. Cloud Sigma does not bundle them together but allows customers to finely tune the combination of resources on demand exactly. Our work focuses on the time granularity and the unit price, aiming to implement an optimized fine-grained pricing scheme with regard to VM-maintenance overhead like VM boot-up cost. For simplicity reasons, the cloud resource bundle mentioned in our paper is referred to as VM instance similar to EC2 on-demand instance.

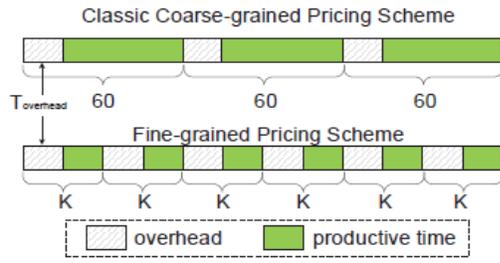
#### **Coarse Grained Model:**

The objective of our work is two-fold, with regard to the classic coarse-grained pricing scheme and inevitable VM-maintenance overhead. On one hand, we aim to derive an acceptable pricing range for both customers and providers, and also derive an optimal price that satisfies both sides with maximized total utility. On the other hand, we hope to find a best-fit billing cycle to maximize the social welfare related to both sides. There are three key terms in our fine-grained pricing scheme: resource bundle, time granularity and unit price. The resource bundle serves as a kind of container to execute task workloads based on user demands. The time granularity is defined as the minimum length in pricing the rented resources. The unit price specifies how much the user needs to pay per time granularity for the resource

consumption. We give an example to illustrate the above terms. In Amazon EC2's pay-as-you-go option, users need to pay \$0:0441 per hour for a small on-demand VM instance (Unix/Linux usage). In this example, the VM instance which bundles CPU, RAM, data storage and bandwidth together is the resource bundle. The time granularity is one hour and the unit price is equal to \$0:044. As another example, in Cloud Sigma, the resource bundle is not an instance but just some type of resource like CPU or RAM. Cloud Sigma does not bundle them together but allows customers to finely tune the combination of resources on demand exactly. Our work focuses on the time granularity and the unit price, aiming to implement an optimized fine-grained pricing scheme with regard to VM-maintenance overhead like VM boot-up cost.

#### **Cloud Time Allocating:**

The designed optimized fine-grained pricing scheme should also satisfy providers, yet providers may suffer higher overhead due to finer pricing rates. In the classic coarse-grained pricing scheme, the provider will suffer To to manage VM instance every one hour service. In the optimized fine-grained pricing scheme, the provider may suffer higher overhead due to more frequent context switch among VM instances. We use an example to highlight the difference between the two pricing schemes in Fig. 3. Obviously, the provider will suffer higher loss of payment in the second pricing scheme with finer granularity due to more frequent VM overheads appearing in the whole service time.



#### 4. ALGORITHM:

**Algorithm 1** Computing maximum price that can be accepted by users

**Input:** The length  $L_u(i)$  of User  $u$ 's all jobs.

**Output:** A vector which contains the maximum prices  $MaxP_{u,K}$  that can be accepted by the user.

- 1: Calculate  $Cost_{u,60}$  for all of the user's jobs with hourly pricing.
- 2: **for** all billing cycles  $K$  **do**
- 3: Calculate the total number of billing cycles for all of  $u$ 's jobs based on the fine-grained pricing.
- 4: Calculate the value of  $MaxP_{u,K}$ .
- 5: **end for**

**Algorithm 2** Maximize Social Welfare

**Input:** The maximum price  $MaxP_{u,K}$  for the user  $u$ ; The minimum price  $MinP_K$  for the provider.

**Output:** The maximum social welfare  $MaxW$  and the best-fit billing cycles  $BF$ .

- 1: Calculate the total cost  $Cost_{u,60}$  for all jobs in the hourly pricing.
- 2: **for** all billing cycles  $K$  **do**
- 3: Calculate the total cost  $Cost_{u,K}$  for all jobs in our optimized fine-grained pricing scheme with optimal price  $\frac{MinP_K + MaxP_{u,K}}{2}$ .
- 4:  $CS_{u,K} \leftarrow \frac{Cost_{u,60} - Cost_{u,K}}{Cost_{u,60}}$ . /\* Cost saving for user  $u$  \*/
- 5:  $RI_{u,K} \leftarrow \frac{Cost_{u,K}/K - Cost_{u,60}/60}{Cost_{u,60}/60}$ . /\* Compute the revenue increment for provider \*/
- 6: **end for**
- 7: **for** all users at billing cycle  $K$  **do**
- 8: Calculate the social welfare  $W_K$  for all users and the provider.
- 9: **end for**
- 10: Select the maximum social welfare  $MaxW = W_{K_m}$ , when the billing cycle is  $K_m$ .
- 11:  $BF \leftarrow K_m$ .

#### 5. CONCLUSIONS

This paper takes the first step to identify and study the partial usage waste issue in cloud computing by analyzing its significance with real-world traces. We propose an optimized fine-grained pricing to solve the partial usage waste issue, with regard to the inevitable VM-maintenance overhead, and find a best-fit billing cycle to maximize the social welfare. By applying the utility theory in economics, we derive an optimal price (the middle point in the range) to satisfy both customers and providers with maximized total utility.

#### REFERENCES

- [1] R. Stefan, K. Holger, M. Pascal, B. Andrew, and L. Mirosław, "Sizing the cloud," Forrester Research, 2011.
- [2] Amazon Elastic Compute Cloud. [Online]. Available: <http://aws.amazon.com/ec2/>
- [3] Google Compute Engine. [Online]. Available: <https://cloud.google.com/pricing/compute-engine>
- [4] E. Deelman, G. Singh, M. Livny, G. B. Berriman, and J. Good, "The cost of doing science on the cloud: the Montage example," in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC), 2008.
- [5] A. Marathe, R. Harris, D. Lowenthal, B. Supinski, B. Rountree, M. Schulz, and X. Yuan, "A comparative study of highperformance computing on the cloud," in Proceedings of the 22nd ACM International Symposium on High Performance Distributed Computing (HPDC), 2013.
- [6] S. Di and C.-L. Wang, "Error-tolerant resource allocation and payment minimization for cloud system," IEEE Transactions on Parallel

and Distributed Systems (TPDS), vol. 24, no. 6, pp. 1097–1106, 2013.

[7] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafir, “The resource-as-a-service (raas) cloud,” in Proceedings of the 4th USENIX conference on Hot Topics in Cloud Computing (HotCloud’12), 2012, pp. 12–12.

[8] CloudSigma. [Online]. Available: <http://www.cloudsigma.com/>

[9] H. Xu and B. Li, “Dynamic cloud pricing for revenue maximization,” *IEEE Transactions on Cloud Computing (TCC)*, vol. 1, no. 2, pp. 158–171, 2013.

[10] S. Di, D. Kondo, and W. Cirne, “Characterization and comparison of cloud versus grid workloads,” in *International Conference on Cluster Computing (CLUSTER)*, 2012, pp. 230–238.

## ABOUT THE AUTHORS

**Mrs. S.JYOTHIREDDY** is pursuing M.Tech degree in, Computer Science and Engineering from Jagruti Institute of Engineering and Technology, Telangana State, India.



**Mr.G.Lakpathi**, presently working as Assistant Professor in, Department of computer science and engineering, Telangana State, India.