Uncertainty-aware Optimization of Resource Provisioning, a Cloud End-user Perspective

Masoumeh Tajvidi, Michael J. Maher and Daryl Essam
School of Engineering and Information Technology, UNSW, Canberra, Australia

Keywords: Cloud Computing, Resource Provisioning, Two-stage Stochastic Programming.

Abstract: Cloud computing offers a customer the possibility of the availability of large computational resources, while paying only for the resources used. However, because of uncertainty in the customers future demand and the future market price for the computational resources, obtaining these resources in a cost-effective and robust way is a difficult problem. The variety of pricing plans is a further complication. In this paper we solve this problem using two-stage stochastic programming, for the first time considering all three available pricing plans, i.e. on-demand, reservation, and spot pricing. Through our experimental implementation, we find that our model can lower the total operational cost by up to 1.5 percent compared to other solutions.

1 INTRODUCTION

Resource provisioning problem in the cloud computing environment can be viewed from different perspectives. the Infrastructure as a Service (IaaS) Provider, the Software as a Service (SaaS) provider, and the cloud end-user. Each trying to maximize their own profit in the resource provisioning plan phase. Due to different goals, requirements, and constraints, the resource provisioning problem in the cloud computing environment needs to be addressed separately, for each stakeholder. In this paper we view this problem from the end-user point of view. The end-user is an individual or an organization, aiming to rent computational resources from a public cloud provider. In cloud computing, provisioning of resources has to be controlled by the end-user, which is a new and unfamiliar paradigm for such users, who are accustomed to working with a fixed set of resources they own. Instead, they encounter a complicated decision making problem of choosing the most suitable type and number of VMs with the best pricing plans, for running their application. There are also uncertain parameters that make this optimization problem even more complicated. Since the resource demand is highly dynamic in nature, its pattern cannot be known, or even be accurately predicted, in advance. Moreover the price of resources varies and is not easily predictable. Numerous cloud providers offer various VM types, namely reservation, on-demand, and spot pricing.

The reservation cost of resources per unit is typically the lowest, but under-provisioning can occur when the reserved resources are unable to fully meet the demand. However, by provisioning more resources with either on-demand or spot plan for extra demand, this problem can be solved, although the user may be charged a higher price. Another potential issue is the over-provisioning problem, which occurs when the reserved resources are more than the actual demand and thence the reserved resources will be underutilized. How to manage all these problems and optimize cost is a critical issue the end-user must deal with.

Although there are plenty research conducted on resource provisioning from the IaaS (Chaisiri et al., 2009) and SaaS (Li et al., 2015) cloud provider's view, There has been little work on solving this problem as it is in the real-world for the cloud end-users. The paper closest to our work optimizes the cost of VM provisioning in the cloud computing environment from the end-user's point of view (Chaisiri et al., 2012) by an optimal cloud resource provisioning algorithm (OCRP). To make an optimal decision, demand and price uncertainty are taken into account. A stochastic integer programming approach with multistage recourse is proposed for optimizing this problem. In this work, however, spot pricing and heteroginity of VMs were ignored. In (Adamuthe et al., 2013), the authors solved this problem from the end-user view in a two phase approach by using genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Heterogeneity of VMs, uncertainty of param-

Tajvidi, M., Maher, M. and Essam, D.
Uncertainty-aware Optimization of Resource Provisioning, a Cloud End-user Perspective.
DOI: 10.5220/0006234103210328
In *Proceedings of the 7th International Conference on Cloud Computing and Services Science (CLOSER 2017)*, pages 293-300
ISBN: 978-989-758-243-1

eters, and existence of various pricing plans were neglected in this paper. In (Genaud and Gossa, 2011), a satisfactory trade-off between cost and speed to process a set of independent jobs is conducted from the end-users' side, but only the on-demand pricing model is taken into account. The main focus of (Zhu and Agrawal, 2010) is an automated and dynamic resource allocation approach, in the cloud environment, based on control theory techniques. An autonomous elasticity controller is proposed in (Ali-Eldin et al., 2012), it changes the number of virtual machine allocated to a service based on both monitored load changes and prediction of future work.

Most of the existing literature, as mentioned above focuses on deterministic formulations over fixed horizons where the scheduler has perfect foresight (Teng and Magoules, 2010). Those that have considered uncertainty in their problem, typically focus on just one aspect (e.g. real-time pricing) (Chaisiri et al., 2012), or use very simple and artificial data (Zafer et al., 2012). Pricing and VM heterogeneity is also neglected in most of them.

In this paper, we propose a mechanism to optimize cost by choosing the most appropriate pricing plans and VM types, while considering the application's demand uncertainty from the consumer side and price uncertainty (on-demand and spot) from the providers side. In order to take into account the uncertainties, we model it as a two stage stochastic optimization problem using the MiniZinc modelling language (Nethercote et al., 2007). The results show that our approach leads to a lower operational cost, compared to previous relevant works.

The rest of the paper is organized as follows. The problem description and formulation is discussed in section II, then in section III is implementation and experimental evaluation, followed by the result discussion in section IV. Finally in section V, conclusions are stated.

2 SYSTEM MODEL AND PROBLEM FORMULATION

The cloud end-user needs to rent a number of VMs for running his application, using public cloud resources with minimum cost. Amazon is a dominant provider in the cloud service market and offers various IaaS and Platform as a Service (PaaS) solutions. The largest and best-known of these is the EC2 IaaS solution (Amazon EC2, 2016). As it is one of the most widely used IaaS providers in both academia and industry, in this work we assume that the end-user wishes to rent resources from the Amazon EC2.

2.1 System Model

Amazon EC2 provides various VM types with the different pricing plans of reservation, on-demand, and spot. Reservation pricing refers to the advance reservation of resources for a specific time period, while securing a lower usage charge. It offers consumers three purchasing variants, "all upfront", "partial upfront", and "no upfront" to purchase reserved instances. With the all-upfront variant, users pay for the entire reserved instance with one upfront payment. This variant provides the largest discount. With the partial-upfront variant, users make a low upfront payment and are then charged a monthly rate for their instances, even for instances that are not utilized in this period of time. The no-upfront variant does not require any upfront payment and provides a monthly rate for the duration of the term. Amazon has recently introduced convertible reserved VMs, which provide customers with additional flexibility to change the VM family, OS, or tenancy, as long as the exchange is for an equal or greater spend on the new convertible reserved VMs. However in this paper we only consider the standard Reservation.

On-Demand pricing lets customers pay for compute capacity by the hour, with no long-term commitments or upfront payments. Depending on the demand of their application, users can simply increase or decrease their compute capacity and only pay for the specified hourly rate for the instances used. Although this pricing model provides convenient flexibility and reliability, it charges customers higher rates than other plans. The on-demand price is not a fixed price and the cloud provider can change it at any time. Spot pricing enables users to bid for unused Amazon EC2 capacity. This price fluctuates periodically, depending on the supply and demand for spot instances. To acquire spot instances, the users place a spot request, specifying the instance type and the maximum price they are willing to pay per hour per instance. If the customer's bid price meets or exceeds the current spot price, the requested instances are granted and they will run until either the user chooses to terminate them or the spot price increases above the maximum bid price. In the latter case, the instances are terminated immediately by the cloud providers with 2 minutes notice. The actual price users pay for their instances is the spot market price, regardless of their bid price. Due to the uncertain availability of spot instances and the potential interruptions they may bring, the spot instance plan is only practical for fault tolerant applications.

When it comes to spot instances, a big challenge is choosing a good bidding strategy. There are various

strategies proposed in the literature(Tang et al., 2012), but generally one can bid high as a means of ensuring to obtain instances with less volatility, or bid lower to optimize costs and send any overflow to on-demand or reserved instances. The most common strategy, however, is to bid on-demand price, called "always bidding on-demand price". With this strategy customers ensure that they will get a discount over on-demand, plus they have a lower chance to be interrupted. In our model we use this simple and effective bidding strategy.

The main issue of the resource provisioning problem for users, is the complexity of dealing with multiple pricing plans, purchasing variants, and VM types, as well as the uncertainty of on-demand and spot prices. As reserved instances are reliable and cost effective resources, they can be considered one of the best options for customers. However users need to decide about them in advance, before running their application, when the real workload and prices are not known. This decision should be made carefully, because there is a long term commitment reserving VMs (at least 1 year). Therefore, more information about future workload can help. Based on this information, a more accurate decision can be made and then, after starting the application, any excess demand can be fulfilled using the more flexible pricing plans of on-demand and spot. As a result, the problem splits into two steps: deciding on the number of reserved VMs, before the uncertain parameters become known, and then compensating for any under-provisioning with on-demand or spot pricing plans.

Having split our uncertain problem into two phases, one of the most appropriate techniques to solve it, is stochastic programming (Birge and Louveaux, 2011). Stochastic programming is an approach for modelling optimization problems that involve uncertainty. When the parameters are uncertain, but assumed to lie in some given set of possible scenarios, a good solution might be one that is feasible for all possible parameter choices and optimizes a given objective function. Stochastic programming models are similar in style, but take advantage of the fact that probability distributions of the uncertain data are known or can be estimated. Often these models apply to settings in which decisions are made repeatedly in essentially the same circumstances, and the objective is to come up with a decision that will, on average, perform well. The most widely applied and studied stochastic programming models are two-stage (linear) programs. Here the decision maker takes some action in the first stage, after which a random event occurs that affects the outcome of the first-stage decision. A recourse decision can then be made in the second stage to compensate for any bad effects that might occur as a result of the first-stage decision(Shapiro and Philpott, 2007).

The first stage decision, therefore, is deciding on the number of reserved instance and the reservation purchasing variants for a one hour time slot, based on an estimation of the future workload. A recourse decision in the second stage is acquiring more VMs on On-demand or spot, if required, in the next hour when the uncertain parameters become known. The detailed explanation of the problem and its formulation is discussed in the following section.

2.2 Problem Formulation

The objective of this problem is to minimize the total cost of provisioning resources. The decision variables are the number of each VM type provisioned under different purchasing variants and pricing plans. See Table 1 for notation. The decision variable x_{ik}^R is the number of reserved VM type i, subscribed to purchasing variant k in the first stage, while x_{ik}^O denotes the number of operating VM type i with purchasing variant k in the second stage. The operating cost is the hourly rate of the reserved VM's utilization cost. Also decision variables x_i^D and x_i^S, respectively, are the number of on-demand and spot VMs of type i in the second stage. We have four provisioning costs, formulated as follows:

- The total Reservation Cost, or the upfront cost of reserving resources, where c_{ik}^R is the price of VM type i with purchasing variant k:
$$C_{ik}^R = \sum_i \sum_k x_{ik}^R c_{ik}^R \quad (1)$$
- The total Operational cost, i.e. the hourly cost of the reserved VMs actually used in the second stage, where c_{ik}^O is the price of VM type i with purchasing variant k:
$$C_{ik}^O = \sum_i \sum_k x_{ik}^O c_{ik}^O \quad (2)$$
- The total On-demand cost, where c_i^D is the price of VM type i:
$$C_i^D = \sum_i x_i^D c_i^D \quad (3)$$
- The total Spot cost, where c_i^S is the price of VM type i:
$$C_i^S = \sum_i x_i^S c_i^S \quad (4)$$

The objective function z is the total expected provisioning cost. It is the main objective function of this problem.

$$Min \;\; z = \sum_i \sum_k c_{ik}^R . x_{ik}^R + IE_\Omega[\Phi(x_{ik}^R, \omega)] \quad (5)$$

Table 1: Notation of the problem.

Symbol	Description
I	set of VM Types
R	set of VM resources or features; CPU and Memory
T	set of Tasks
Cap_i^{CPU}	CPU Capacity of VM type i
Cap_i^{Memory}	Memory Capacity of VM type i
Req_t^{Memory}	Amount of memory required for completing task t
Req_t^{CPU}	Amount of CPU required for completing task t
S	set of scenarios
K	Set of reservation purchasing variants, all-, partial-, or no-upfront
$MaxBudget$	User's maximum budget
C_{ik}^R	Reservation Cost of VM type i subscribed purchasing variant k
x_{ik}^R	Number of reserved Vms type i subscribed purchasing variant k
C_{ik}^O	Operation Cost of VM type i, subscribed purchasing variant k
x_{ik}^O	Number of Operation VM type i, subscribed purchasing variant k
C_i^D	On-demand Cost of VM type i
x_i^D	Number of on-demand VM type i
C_i^S	Spot Cost of VM type i
x_i^S	Number of Spot VM type i

subject to:

$$x_{ik}^R \in \mathbb{N} \tag{6}$$

Where $IE_\Omega[\Phi(x_{ik}^R, \omega)]$ is the expected cost under uncertainty Ω, and Φ is the recourse optimization problem. The objective of $\Phi(x_{ik}^R, \omega)$ is to minimize the cost under uncertainty given scenario ω:

$$Minimize \sum_{\omega}(x_{ik}^O c_{ik}^O + x_i^D c_i^D + x_i^S c_i^S) \tag{7}$$

subject to:

$$x_{ik}^O \leq x_{ik}^R \tag{8}$$

$$z \leq MaxBudget \tag{9}$$

$$TotalCPU \geq \sum_t Req_t^{CPU} \tag{10}$$

$$TotalMemory \geq \sum_t Req_t^{Memory} \tag{11}$$

The first constraint (8), limits the number of operating reserved VMs (x_{ik}^O) of each type to be less than or equal to the number of reserved VMs (x_{ik}^R). As discussed earlier, the customer reserves a number of VMs in the first stage and pays an upfront fee for them, then in the second stage these reserved VMs can be used by the customer. So the number of used VMs (operating VMs) in the second stage cannot be greater than the number of reserved VMs in the first stage. The second constraint (9) states that the total provisioning cost, or the objective function z, cannot be greater than the maximum budget the customer specified for running their application.

Constraints (10) and (11) both ensure that the amount of required resources for all tasks is satisfied by the VMs acquired in the second stage. The instance types comprise varying combinations of CPU and memory, and give the customer the flexibility to choose the appropriate mix of resources for their applications. Therefore, each task can be run on multiple VMs simultaneously, just as each VM can host multiple tasks of a particular application. $TotalCPU$ and $TotalMemory$ are the available CPU and Memory in the second stage, and are defined as follows:

$$\begin{aligned} TotalCPU = &\sum_{ki} x_{ik}^O Cap_i^{CPU} \\ &+ \sum_i (x_i^D + x_i^S) Cap_i^{CPU} \end{aligned} \tag{12}$$

$$\begin{aligned} TotalMemory = &\sum_{ki} x_{ik}^O Cap_i^{Memory} \\ &+ \sum_i (x_i^D + x_i^S) Cap_i^{Memory} \end{aligned} \tag{13}$$

where Cap_i^{CPU} and Cap_i^{Memory} are the CPU and memory capacity of VM type i, specified by the cloud provider.

3 EXPERIMENTAL EVALUATION

3.1 Data Set

Two main characteristic of cloud workloads, relevant to our work, are the job length and the resource utilization (Di et al., 2012). In order to have a representative data set for cloud, the jobs should finish quickly, on the order of a few minutes: 55% of tasks in cloud finish within 10 minutes and about 90% of tasks' lengths are shorter than 1 hour(Di et al., 2012). In addition the jobs usually have low resource demand for CPU and memory, as they are mostly interactive and real-time, in contrast to scientific batch jobs (Di et al., 2012).

Therefore, for evaluating the proposed approach under realistic and various working conditions, we found workload traces of the DAS-2 multi-cluster system (DAS2, 2009) well-suited for our experiments. This data set were obtained from Parallel Workload Archive (Parallel Workload Archive, 2016) and consist of over two hundred thousand small jobs with short run-time period.

These data are real users' request logs collected from five different universities in the Netherlands. We use twelve-month workloads on DAS-2 clusters in the year 2003. The main characteristics of the data used in this work include: Number of Allocated Processors, Used Memory, User ID and Group ID fields. 12 user groups with various jobs (different CPU and Memory usage pattern) are extracted from the data set. As our problem is viewed from the end-user's perspective, we use top 12 dominant users' groups (in terms of number of tasks) individually for evaluating our work, further discussion on group behaviours are made in the following sections.

The other data required for our experimental evaluation is the VM prices for all three pricing plans. The VM prices of the prevalent IaaS cloud provider, Amazon EC2 (Amazon EC2, 2016), i.e., the standard reserved, operating, on-demand, and spot prices were extracted from Amazon EC2s official website (Amazon EC2, 2016) over May 2016.

3.2 MiniZinc Model

The model was implemented in the MiniZinc modeling language (Nethercote et al., 2007), using the G12 MIP solverWe model the problem in two separate MiniZinc models. The first model solves the problem of determining the number of reserved VM, before the unknown parameters become known. The workload probability distributions for the user group are extracted from the DAS-2 data set (DAS2, 2009).

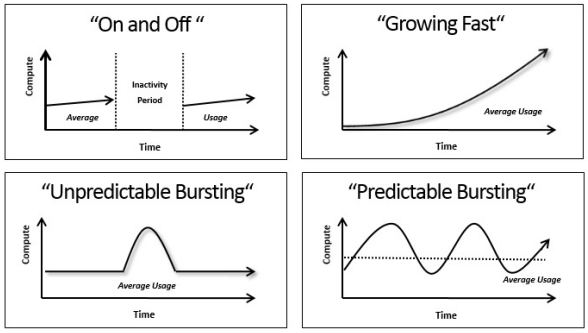


Figure 1: Workload patterns in cloud computing (Steve Clayton, 2009).

To represent the uncertainty of parameters, 50 workload scenarios are chosen pseudo-randomly with a uniform distribution from that range. Similarly, 50 cost scenarios for on-demand and spot prices were generated based on the extracted data of May 2016. The outcome of this model is the optimum number of reserved VMs, and the expected total cost based on the 50 scenarios.

In the second stage, the real workload and real prices of on-demand and spot VMs become known. A single set of workload demand is again pseudo-randomly generated from our DAS-2 data set(DAS2, 2009) and is used with the prices captured from Amazon EC2 (Amazon EC2, 2016). Based on these data and determined number of reserved VMs from the first model, the outcome of the stage 2 model is the optimal number of spot and on-demand instances as well as the actual total cost.

The model contains some approximations. In general, not all instances of a particular type perform to exactly the same standards, because of variations in the physical hardware that is allocated for them and possible multi tenancy (Mao and Humphrey, 2012). We use the minimum guaranteed performance of EC2 instances as the baseline to ensure that sufficient resources are available for each job.

We repeated the first and second stage models 20 times, for each individual group independently and for different options, that are described in section 3.4, to see how total price varies. They all share the same data and configuration during each run and the overall MiniZinc execution time for each run is less than 1 minute.

3.3 Cloud Workload Patterns

Microsoft research (Steve Clayton, 2009) identified four workload patterns as most suitable for cloud computing applications, as shown in Fig 1. The workloads are:

On and off: Applications that have a relatively short period of activity and after producing a result they can be terminated (such as image processing applications).

Predictable/unpredictable Bursting: In predictable bursting workload, the peak demand period is roughly known. In unpredictable bursting workload, however, it is unknown when the peak demand will be.

Growing Fast: This is the scenario of start-up companies, when suddenly their services become successful and the infrastructure they are running their application on cannot cope.

3.4 Pricing Plan Options

We repeat the experiment for four different options that are various combinations of Amazon EC2 pricing plans. The options are:

- Reservation-Ondemand-Spot (ROS) option: This is our main option; it considers all three pricing plans. The ROS option is ideal for cloud users with a "predictable bursting" workload. As we will discuss later in the result section, a better prediction of workload can lead to a more cost saving due to more accurate decisions made in the first stage on number of reserved VMs, and then extra demand can be optimally satisfied with on-demand and spot instances, if the application is fault tolerant enough to deal with spot.
- Reservation-Ondemand (RO) option: Only reservation and on-demand pricing are considered for this option, it corresponds to the model of (Chaisiri et al., 2012). When the user's application is time critical and deadline sensitive, the spot instances cannot help much. So even if they have the ideal "bursting" workload, they cannot consider ROS. Instead they can apply the RO option.
- Ondemand-Spot (OS) option: The OS (no reservation) option only considers on-demand and spot pricing plans.
- Ondemand (O) option: In the O (all on-demand) option, the only pricing plan is the on-demand plan. With an application with "on and off" demand that is deadline sensitive, the user has no choice but the expensive on-demand instances.

Each option has its place and when used in the correct manner savings can be made.

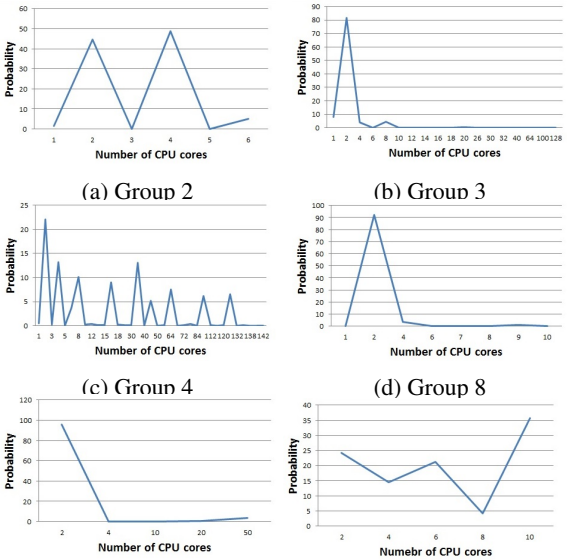


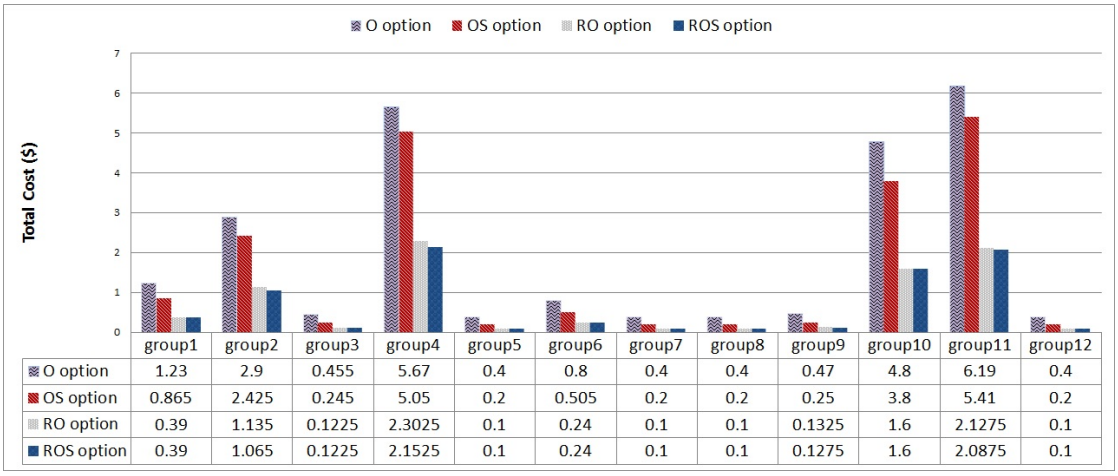
Figure 2: CPU usage probability distribution.

4 RESULTS

Figure 3 shows the average total cost of all groups for all the above mentioned options. The ROS achieves the lowest total cost for all 12 groups, while the O yields the highest total cost due to the fact that on-demand instances are the most expensive ones. The next highest cost belongs to the OS and the RO options, respectively.

Comparing the total cost of the ROS to other options in average, there is a 70% and 53% cost saving over the O and the OS options, respectively. Also there is an average of 1.53% cost reduction by using the ROS option rather that the RO. In some groups, the total costs of ROS and RO options are the same, such as groups 1, 3, 5, and 8, where spot instances are not needed. Whereas for groups 2, 4, 9, and 11 there is a 6.6%, 7%, 4%, and 2% cost saving, respectively, by using the ROS option. Therefore the ROS option can be more or less beneficial for users with different demand attributes.

To get an idea of the group's demand attributes, Figure 2 shows the probability distribution for jobs' CPU demand for 6 different user groups. Groups 3 and 8 have almost similar distribution patterns, their density of the number of CPU demand is mostly concentrated around a specific range (1 to 4 CPU core). Other illustrated groups, group 2, 4, 9, and 11 have different probability distribution patterns, however they all have one thing in common: a large variety in the number of CPU cores required.



	group1	group2	group3	group4	group5	group6	group7	group8	group9	group10	group11	group12
O option	1.23	2.9	0.455	5.67	0.4	0.8	0.4	0.4	0.47	4.8	6.19	0.4
OS option	0.865	2.425	0.245	5.05	0.2	0.505	0.2	0.2	0.25	3.8	5.41	0.2
RO option	0.39	1.135	0.1225	2.3025	0.1	0.24	0.1	0.1	0.1325	1.6	2.1275	0.1
ROS option	0.39	1.065	0.1225	2.1525	0.1	0.24	0.1	0.1	0.1275	1.6	2.0875	0.1

Figure 3: Total Cost Comparison for all on-demand (O), no reservation (OS), no spot (RO), and the complete ROS options.

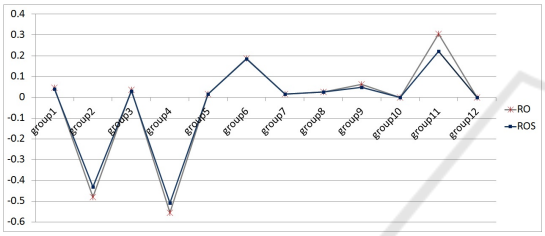


Figure 4: The difference (expected cost − actual cost) in $ for the ROS and RO options.

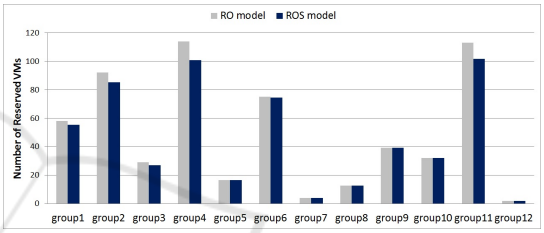


Figure 5: Number of Reserved VMs in ROS and RO options, average over 20 runs.

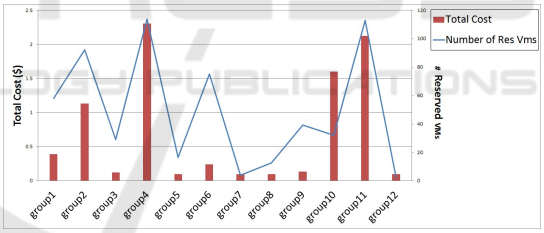


Figure 6: Total cost and number of reserved VMs in the RO option.

Consequently, we find that the probability distribution of CPU and Memory usage is a factor in cost reduction by ROS, compared to RO. A more narrow demand distribution results in less cost saving, whereas a wider range of job requirements leads to a better cost saving. This is true for all groups, even for the ones that are not shown in Figure 2. The underlying reason appears to be that the workload of a concentrated distribution is more predictable, as the first stage decision is made based on a set of workload scenarios that are more likely to be similar to the real time workload of the second stage. That is why the decision in the first stage can be made more accurately by the solver, and therefore there is less need to handle jobs in the second stage, and in particular, less need to employ spot instances.

The accuracy of the first-stage decision can be calculated as the the difference between expected cost in the first stage and the actual cost in the second stage. The closer to zero this number is, the more accurate was the decision for the number of reserved VMs in the first stage. Figure 4 shows the difference of total expected and total actual cost for the ROS and OS options. In terms of overall accuracy on average, the accuracy of the ROS is 5% better than the RO, while for groups 2, 4, 9, and 11 this number is more significant. Accordingly, the total cost saving for them in the ROS is higher.

As depicted in Figure 5, for all groups, the number of reserved VMs with the RO option is greater than or equal to the number of reserved VMs with ROS, 4% greater on average. When there is not a low cost spot VM available, the solver reserves more VMs in the first stage, to avoid paying for expensive on-demand VMs later in the second stage. These extra reserved VMs may not be used in the second stage, therefore the user may pay for unused reserved instances. Hence, the number of reserved VMs determined in the first stage has an important impact on the total price and leads to a higher price for the RO model, as shown in Figure 6.

Generally, the total cost of our main option, ROS, is the lowest cost among all other options, but specif-

ically it leads to more cost savings when the CPU demand is more widely distributed or is less predictable. The 1.5 percent cost saving by using spot instances can be translated to a significant amount of money when a huge amount of resources is rented for the cloud provider by a particular company. So by choosing a cloud provider that offers spot instances, users can benefit from the potential huge discounts.

5 CONCLUSIONS

The resource provisioning problem in the cloud environment is viewed from the end-user perspective, in this paper. While there are uncertainties in the parameters and heterogeneity in VM type and prices, the optimal number of VMs are determined using a two stage stochastic optimization problem.

Our results show that in general, the proposed ROS option is cheaper than the others. We saw that user groups with almost uniform CPU requirements do not pay a penalty when using the RO option, but other user groups can pay a significant penalty. Other options are significantly more expensive. Overall, our results demonstrate and quantify the effects that job mix and workload patterns can have on the resource provisioning cost for cloud end-users.

We plan to extend this study by evaluating our model with a real cloud dataset (Google Cluster Dataset) and improve our model to a multi-stage stochastic model for getting more precise results, and also make the model more flexible by considering the convertible reserved VMs.

REFERENCES

Adamuthe, A. C., Bhise, V. K., and Thampi, G. (2013). Solving resource provisioning in cloud using GAs and PSO. In *Engineering (NUiCONE), 2013 Nirma University International Conference on*, pages 1–5. IEEE.

Ali-Eldin, A., Kihl, M., Tordsson, J., and Elmroth, E. (2012). Efficient provisioning of bursty scientific workloads on the cloud using adaptive elasticity control. In *Proceedings of the 3rd workshop on Scientific Cloud Computing Date*, pages 31–40. ACM.

Amazon EC2 (2016). Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/.

Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

Chaisiri, S., Lee, B.-S., and Niyato, D. (2009). Optimal virtual machine placement across multiple cloud providers. In *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific*, pages 103–110. IEEE.

Chaisiri, S., Lee, B.-S., and Niyato, D. (2012). Optimization of resource provisioning cost in cloud computing. *Services Computing, IEEE Transactions on*, 5(2):164–177.

DAS2 (2009). The Distributed ASCI Supercomputer 2. http://www.cs.vu.nl/das2/.

Di, S., Kondo, D., and Cirne, W. (2012). Characterization and comparison of cloud versus grid workloads. In *2012 IEEE International Conference on Cluster Computing*, pages 230–238. IEEE.

Genaud, S. and Gossa, J. (2011). Cost-wait trade-offs in client-side resource provisioning with elastic clouds. In *Cloud computing (CLOUD), 2011 IEEE international conference on*, pages 1–8. IEEE.

Li, S., Zhou, Y., Jiao, L., Yan, X., Wang, X., and Lyu, M. R.-T. (2015). Towards operational cost minimization in hybrid clouds for dynamic resource provisioning with delay-aware optimization. *Services Computing, IEEE Transactions on*, 8(3):398–409.

Mao, M. and Humphrey, M. (2012). A performance study on the vm startup time in the cloud. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 423–430. IEEE.

Nethercote, N., Stuckey, P. J., Becket, R., Brand, S., Duck, G. J., and Tack, G. (2007). Minizinc: Towards a standard CP modelling language. In *Proc. Int. Conf. on Principles and Practice of Constraint Programming*, pages 529–543.

Parallel Workload Archive (2016). Logs of Real Parallel Workloads from Production Systems. http://www.cs.huji.ac.il/labs/parallel/workload/logs.html.

Shapiro, A. and Philpott, A. (2007). A tutorial on stochastic programming. *Manuscript. Available at www2. isye. gatech. edu/ashapiro/publications. html*, 17.

Steve Clayton (2009). MSDN blog. https://blogs.msdn.microsoft.com/stevecla01/2009/11/26/optimal-workloads-for-the-cloud/.

Tang, S., Yuan, J., and Li, X.-Y. (2012). Towards optimal bidding strategy for Amazon EC2 cloud spot instance. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 91–98. IEEE.

Teng, F. and Magoules, F. (2010). Resource pricing and equilibrium allocation policy in cloud computing. In *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, pages 195–202. IEEE.

Zafer, M., Song, Y., and Lee, K.-W. (2012). Optimal bids for spot vms in a cloud for deadline constrained jobs. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 75–82. IEEE.

Zhu, Q. and Agrawal, G. (2010). Resource provisioning with budget constraints for adaptive applications in cloud environments. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, pages 304–307. ACM.