A Cloud Bidding Framework for Deadline Constrained Jobs

H M Dipu Kabir[1], Abadhan S. Sabyasachi[2], Abbas Khosravi[1], M Anwar Hosen[1],
Saeid Nahavandi[1], Rajkumar Buyya[3]
[1]Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University, Australia
[2]Department of CSE, The Hong Kong University of Science and Technology, Hong Kong.
[3]Cloud Computing and Distributed Systems (CLOUDS) Laboratory,
Department of Computing and Information Systems The University of Melbourne Melbourne Victoria Australia
{dkabir, abbas.khosravi, anwar.hosen, saeid.nahavandi}@deakin.edu.au, abadhan@cse.ust.hk, rbuyya@unimelb.edu.au

***Abstract*—Reliable completion of the computing jobs through Amazon spot instances (SIs) with proper bargaining is challenging. Therefore, an SI bidding system is developed for deadline constrained jobs considering both the conditions of the market and the condition of the user. The system tries to bargain with the provider by bidding low when the task is not urgent. After that, the system increases the price or the price distribution gradually when the progress is lower than required. To calculate the bid distribution, we compute the probability density of the price after five minutes. Then, we apply our developed equations to compute bid-prices from the probability density function. Equations are easily interpretable to both humans and machines. We also consider long-term probability distributions of the prices for the reliable completion of the job. Tasks with several days deadline are prescribed to bid considering the daily price-curve. According to the evaluation of Amazon SI price, the proposed system effectively saves 79%-87% for jobs with several hours deadline and saves 82%-100% for jobs with several days deadline compared to the on-demand instances. Moreover, our algorithm helps all bidders by keeping the price low.**

***Keywords*—Amazon EC2, Spot Instance Management, Probability Density, Truthful Bidding, Cloud Bargaining.**

I. Introduction

Computation extensive jobs are increasingly being executed on public cloud platforms such as the Amazon Elastic Compute Cloud (Amazon EC2) due to several advantages. These include reliability, security, zero maintenance, meeting variable demand, the payment for usage only etc. In order to meet the peak demand of on-demand and reserved instances reliably, Amazon has installed abundant EC2 instances which have resulted in a large number of unused EC2 instances. Amazon has developed an auction-based selling system and through the system, unused instances, known as spot instances (SIs) can be used at a much lower price most of the time [1], [2]. The auction system of Amazon is similar to the uniform price auction or the clearing price auction [3] that involved several uncertainties; such as- variable availability of SI and it is not stated whether providers are looking for the profit maximization or the resource utilization [4].

Although the spot instances (SIs) are failure prone, their price is usually 80-90% lower compared to the on-demand instances [5]–[9]. That cost efficiency brings the interest of many researchers in cloud computing and economics; they develop algorithms for the efficient use of SIs. Through efficient system development with the help of checkpointing [10] and bidding [11] strategies, researchers have executed deadline constrained jobs with a satisfactory quality of services (QoS) [12]. Researchers are also aiming to finish 80-90% of their job within the soft deadline and also, trying to finish the rest within the 105% time of the soft deadline. Although the QoS is slightly degraded with these systems, they are helping others with the bargaining [13] and keeping the price lower [14]. Many jobs do not have any hard deadline, such as scientific computing for research; but a good turnaround time with cost efficiency is expected and researchers are also developing algorithms for them [15]. In order to help algorithm development groups and skilled bidders, some research only characterizes the pricing of different SIs [16], [17]. In summary, the aim of everyone is to perform the cost minimization [18]–[20] through SIs with a desirable QoS.

II. Background

A. Importance of the Truthful Bidding: A Numerical Example

Let us consider three bidders bids for two tickets. They follow rules of the uniform price auction system with the lowest winning bid payment method. Their bids are \$1, \$2 & \$5 respectively and the ticket seller wants to maximize his profit without considering the resource utilization. Therefore, he sets the price at \$5 and only the third bidder gets the ticket. As a result, one ticket is wasted. Moreover, if the third user bids higher only to ensure the ticket and his utility for the ticket is lower than \$5, he is penalized with the price.

Although the Amazon EC2 SI bidding system considers a capacity function to provide some incentives to users, the weight of incentives can be low. In addition, the average SI price is roughly 10-20% of the on-demand price and the maximum limit for the bid is 10 times the on-demand price [1]. As a result, the bidding of 1-2% careless bidders can potentially raise the spot price to a value 10 times that of the on-demand price. In a documented situation, the spot price rose to \$999.99 per hour where the average price was about \$0.44 per hour [21]. That happened due to a few careless bidders bidding at that high price to ensure the continuous availability of SIs with possible price discounts. In such a situation, all bidders culpable or not suffers.

978-1-5386-6376-9/19/$31.00 ©2019 IEEE

B. Motivation towards the Bidding Strategy Development

Existing bidding strategies are relying on the point prediction which is a value corresponds to the minimum error (RMSE, MAPE etc.) [22]–[25]. Bidding at the point prediction provides roughly a 50% probability of winning the bid. In addition, an interruption may occur in the middle. Therefore, spot instances (SIs) are not reliable to execute urgent tasks. However, the task is accomplishable when the time to the deadline is much higher compared to the required completion time. Moreover, users may get free partial hours and the payment gets reduced. Therefore, both the urgency of the task and the bid predictions with different assurances of winning the bid are needed.

C. Probable Optimization Function of Amazon

Although Amazon has not disclosed the optimization functions for the price on their website, several research groups have developed optimization functions based on the historical data. The optimization function consists of two major parts [26], commonly known as revenue maximization and capacity utilization [27], [28]. The revenue maximization function is the multiplication of the number of accepted SIs and the price of SI. In order to increase the user-friendliness of the EC2 bidding system, Amazon is also considering a capacity maximization function. Capacity optimization function increases logarithmically with the increment of the number of accepted bids. Equation (1) presents the profit function, equation (2) presents the capacity function and the Amazon EC2 SI provider's probable optimization function is the maximization of the sum of these functions, presented as equation (3).

$$Profit\ Function = \pi(t)N(t) \tag{1}$$

$$Capacity\ Function = log(1 + N(t)) \tag{2}$$

$$\max_{\pi(t)} \quad \pi(t)N(t) + \beta log(1 + N(t)) \tag{3}$$

where, $N(t)$ is the number of accepted SIs, $\pi(t)$ be the price per accepted SI, and β is the weight of the utilization term.

The following equation visualizes the subtle difference in price and the number of users:

$$\max_{\pi(t)} \quad (\pi(t)+\Delta\pi)(N(t)-\Delta N)+\beta log(1+N(t)-\Delta N) \tag{4}$$

Here, the number of users is decreased by ΔN due to $\Delta\pi$ price increase.

Although the capacity utilization function is embedded with the price optimization equation to reduce unexpected terminations, it does not reflect the user-friendliness when a large number of users is already available; when $\Delta N < N(t)/10$. However, when the number of accepted bids is small compared to the change $\Delta N > N(t)$, the capacity function dominates the optimization equation.

In summary, the optimization equation tries to increase the number of accepted bids when the number of accepted bids is too low but the equation only maximizes the profit when a large number of bids are already accepted.

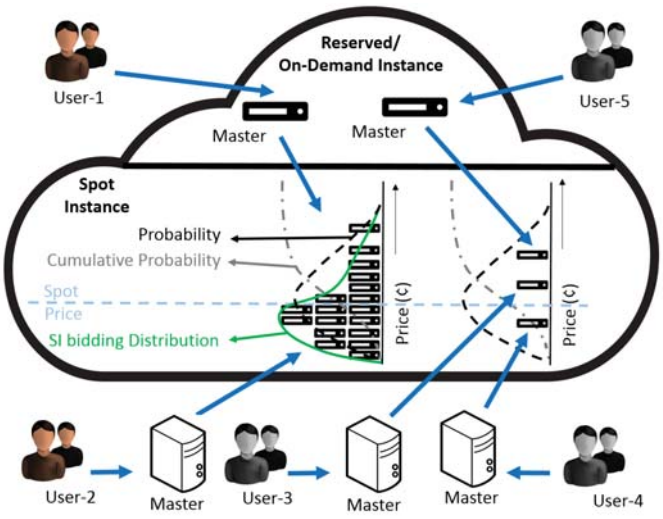

Fig. 1. Bidding strategy for different users. User-1 and user-2 are the type-1 users; one is using a reliable cloud instance as a master. User-3, user-4, and user-5 are type-2 users. They can use different masters and the value of their bid depends on how close the deadline is.

III. SI Bidding Strategy Based on Users Demand and Market Condition

The provider's probable optimization formula is presented in the previous section (Section II). An individual user can not change the formula but he can bargain by submitting bids intelligently. This section presents an efficient bidding strategy of the user that can bring both profit maximization and capacity maximization at a slightly lower price compared to the expected spot price.

A. Who can Bargain through the Bidding

The users do not have any permanent server or a low configuration computing machine for monitoring bids cannot go towards the risky SI bidding. A user bids at the on-demand value may also lose access to the server during the execution [29]. Jobs with switching masters eventually terminate and require relaunching because it can also potentially happen that servers of all configurations are claiming more than the on-demand value. When there is no computing machine left for re-bidding and re-launching the job, human involvement requires for the re-launching. As humans cannot work for 24-hours, there is a delay for the re-launching and the progress of the task suffer until the re-launch time. Also, bidding at a too high price can potentially harm all bidders [30].

The base server or the low configuration bidding and path forwarding machine is the master or the master of masters. That machine can be a physical computing machine, owned by the user or it can be an on-demand or a reserved instance in the cloud. Through the permanent computing machine, the users can bid for spot instances (SI). Based on the nature of their jobs, we classify biddable users into two categories-

1) *Users with parallelizable tasks*
The users' task needs to be large enough to place multiple bids. The user needs to distribute bidding prices so that the providers' optimization occurs at

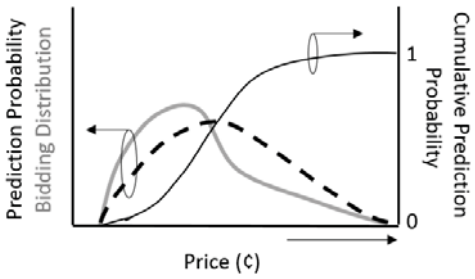

Fig. 2. A rough sketch of the prediction probability (black dashed line), the cumulative probability distribution (narrow black line) and the bidding distribution (gray line).

a slightly lower price. In figure 1, user-1 and 2 are type-1 users.

2) *Users with nonparallelizable tasks*
Although the user cannot set a number of bids, he can still bid intelligently. However, the user needs a longer deadline to finish the job to bargain. In figure 1, user-3, 4 and 5 are type-2 users. When the remaining parallelizable job is accomplishable in an SI within one hour, the user may consider himself as a type-2 user to avoid overbooking.

In figure 1, user-2, 3 and 4 use personal computing machine and user-1 and 5 use a reliable cloud instance as a master.

B. Bidding Strategy for the Type 1 User

1) Graphical Explanation: Firstly, the bidding strategy is graphically presented in figure 2 for the ease of readers. The providers' optimization formula sets a price which aims the maximization of benefits with little tolerance for the social welfare. The value, $Profit\ Function$ is the multiplication of the number of users and profit per user. The social welfare value increases logarithmically with the number of active users of the corresponding SI. Both of the social welfare and the profit maximization are maximized at a slightly lower SI price when a large number of users are bidding at a slightly lower price. Therefore, we propose the providers' optimization function presented as the equation (3). Black dashed curve in figure (2) represents a rough sketch presenting the probability distribution of an SI price. The thinner black curve represents the cumulative distribution of probability and the gray curve represents a wise bidding distribution from the bidders' perspective.

Type-1 users should also bid for all of their jobs. If one user bids for half of the spot instance for the current hour and keeps half instances for the next hour, his influences the auction market poorly. Therefore, bidding for more instances is better unless it is an overbooking.

2) Equation for Automated Bidding: Although the graphical representation in Fig. 2 is convenient to human, an equation is handy for automated machines to generate the bidding distribution from the probability distribution. Eqn. 5 presents a formulation for calculating the bidding distribution from the cumulative probability distribution of price.

$$i^{th}\ bid = C_{pp}^{-1}(\{\frac{i}{N_{SI}+1}\}^n) \quad (5)$$

$$n = 2 \times (1 - \frac{1}{\eta \times T_D}) \quad (6)$$

were, $i^{th}\ bid$ is the price of i^{th} server of the bidding system, $C_{PP}(Price)$ means the percent cumulative prediction probability from 0 to that $Price$ and $C_{PP}^{-1}(x)$ means the price at the $100x$ percent cumulative prediction probability. N_{SI} is the number of servers currently bidding and n is the skew factor, T_D is the deadline in full hour.

The value $\frac{i}{N_{SI}+1}$ is always lower than 1 and any power $(n > 1)$ of that value is less than this value and the bid distribution in equation (5) becomes lower than the probability distribution. The skew factor $n \approx 2$ set when the deadline is more than several hours and the value decreases over time, following the equation (6). The fitting parameter η is kept 1, but users can change it depending on the situation and their priority.

Users, working with a large number of SI can increase i by a certain number a and set the bid of the next a instances with the same value, calculated by the equation (5).

3) Bidding Again or Not: When a sufficient number of bids are accepted, the bidder needs to keep bids of lower prices. That can potentially bring a price reduction. When running SIs accomplish a significant portion of the task, the user does not need all of the bids for the next hour. Therefore, he needs to remove a few unaccepted bids of higher prices. However, when the user fails to win sufficient bids, he needs to remove some of the unaccepted bids of lower prices and re-bid maintaining the new bid distribution after five minutes. Bidding by following the curve, some of those bids become successful even when the price is higher than the mean value of the probability distribution. Therefore, the user finishes a portion of his task when the price is much higher than the expected [31], [32].

4) Calling On-demand Instances During the Urgency: The user needs to consider the adverse scenarios to complete jobs reliably. The deadline can be very close to the minimum time, required to finish the job. In such situation, the user may not rely on SIs. He needs to buy one on-demand instance. However, the probability of happening such situation is quite low. Suppose the user got 4 hours for completing the job and therefore, he bids for jobs with n=1.5, probabilistically 37% bids are accepted. When the acceptance rate is low, he continuously closes open instances and re-bids with 5-minutes interval. A few tasks or no task wait for the final hour [33].

C. Bidding Strategy for the Type 2 User

Type-2 users can bid for only one instance. If they are bidding for two instances and winning both of the bids, one instance becomes a waste. Therefore, type-2 users usually bid for a single instance at a slightly higher price to ensure the win of the bid. However, if a large number of type-2 users bid at a higher price, the price can potentially hike following the providers' optimization formula. To bargain with the provider through the bidding of one instance, the user can follow the equation (7).

$$Price_{bid} = C_{PP}^{-1}(min\{\frac{T_R + T_I + T_{IN} + T_S}{T_D} + \theta, 1\}) \quad (7)$$

were, $Price_{bid}$ is the bidding price. $C_{PP}(Price)$ means the percent cumulative prediction probability from 0 to that $Price$

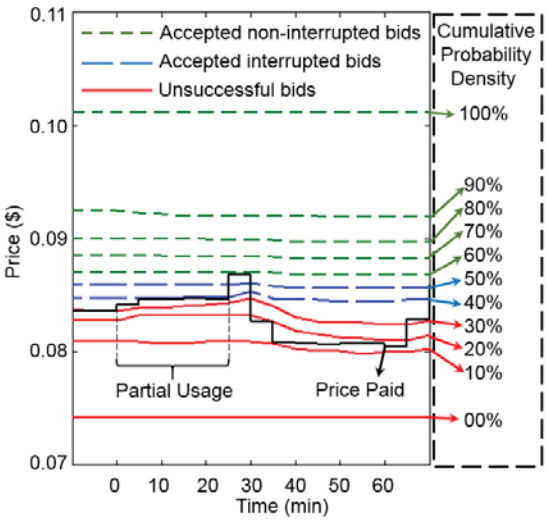

Fig. 3. The method of determining the performance; bid success, interruption, price etc.

and $C_{PP}^{-1}(x)$ means the price at $100x$ percent cumulative prediction probability. T_R is the time required to complete the remaining job; T_I is the initialization time. Figure 2 can provide the readers an easier understanding about $C_{PP}(Price)$ and $C_{PP}^{-1}(x)$ functions. T_{IN} is the interval at which the progress is saved; T_D is the time until the deadline; T_S delay due to each saving operation. Finally a margin θ is kept in order to ensure that there is a sufficient gap between $T_R+T_I+T_{IN}$ and T_D. Keeping a large θ ($\theta > 5\%$) reduce the probability of calling an on-demand server. However, keeping a too large θ can potentially destroy the bargaining. When ($\theta > 50\%$), the users always bids at a higher price compared to the point prediction. In our previous experiment [34], the value of θ is kept $\theta = 10\%$. However, by doing so, the lowest 10% part of the cumulative probability remains out of the feasible bidding region. Therefore the equation 8 is introduced for the calculation of theta. In this equation, the skew factor $n = 2$ for normal conditions and a large negative power of 2 is close to 0.

$$\theta = n^{-\frac{T_D}{T_R+T_I+T_{IN}+T_S}} \tag{8}$$

According to equation 8, the value of margin (θ) is close to zero when the deadline (T_D) is much higher than the time, required for completing the job. When the deadline becomes 3 times higher than the sum of ($T_R + T_I + T_{IN} + T_S$), the value of theta becomes 12.5% and theta becomes 25% and 50% for $T_D/(T_R+T_I+T_{IN}+T_S)$= 2 and 1 respectively. In fact, when the ratio becomes less than or equal to 2, the user starts to bid at 100% cumulative probability distribution value and the user needs to go for the on-demand instance when the ratio becomes 1. As a result, the user bargains based on both the urgency of his task and the condition of the market.

D. Probability Density Calculation for SI Bidding

Currently-available SI bidding techniques mostly use the point prediction or a sufficiently high value that ensures the availability of the instance for a certain amount of time [22]. However, both techniques have several limitations. The point prediction is a value corresponds to the minimum error (RMSE, MAPE etc.) and the value is close to the median. There is roughly a 50% probability that the price is lower than the point prediction [35]. Point prediction has no relation to the users' urgency of the task. The bidders do not have sufficient amount of time to complete the task cannot rely on the point prediction. The point prediction does not contain any information about the heteroscedastic uncertainty [36]–[39]. Also, bidding at a significantly higher price results in the absence of negotiation. That can potentially bring the provider's optimization at a higher price and the provider can potentially maximize the revenue with a fewer number of customers without considering unused instances and waiting customers. That is why probability density based bidding strategies are developed, those can negotiate with the provider depending on the condition of the market and the urgency of the bidder. The method [40] of constructing probability density through historical similarities is applied to evaluate the result.

IV. Performance Evaluation

As we design the proposed algorithm in such a way that everyone gets the benefit when everyone follows the approach. We can not observe the real advantages when others are not using the algorithm. On the other hand, no algorithm become popular unless a few individuals get benefits by following it. We evaluate the performance with the help of existing Amazon trace of one high configuration ($c4.4xlarge\ Linux$) and one lower configuration ($c4.2xlarge\ Linux$) servers. Figure 3 presents the spot price of a typical hour and cumulative probability distribution curves from ($c4.2xlarge\ Linux$) trace. Bids at lower than or equal to 30% cumulative probability values fail. Bids at 40% and 50% values are interrupted after approximately 25 minutes and bids at higher values progress without interruption. Random points are picked from the curves to evaluate performance matrices; such as- bid success, interruption, and price.

A. Performance Evaluation of Jobs with Shorter Deadline (<24 hr.)

Results of bidding at different cumulative probabilities of the price after 5 minutes are analyzed from Amazon traces are presented as Table-I and Table-II. The average results are similar to the results of conventional techniques, resulting in about 85% savings on average compared to the on-demand prices. In addition, the proposed method helps all users by keeping the price lower. However, extreme bargaining degrades the quality of service (QoS) through bid-failure and unexpected terminations. Thus our result analysis section considers two major concerns- the QoS and the cost efficiency.

1) Quality of Service (QoS): A certain number of dedicated servers can provide the optimum QoS. However, unreliable spot instances are selected due to the cost efficiency and that results in a slightly degraded QoS. An acceptable degradation of QoS depends on a certain amount of bid acceptance and less-interrupted service of the instance. When an instance is terminated due to the price increase, the job requires saving the progress during an unexpected termination. Due to these facts, the percentage of successful bids, the percentage of interruption, and the average availability are computed as indicators of QoS.

TABLE I.
ANALYSIS OF BIDDING AT DIFFERENT CUMULATIVE PROBABILITY ON $c4.4xlarge$ $Linux$ SPOT INSTANCES OF US EAST (N. VIRGINIA)

Cumulative Probability	Successful Bids (%)	Interr-uption (%)	Average Lifetime (minutes)	Effective Price (per hr.)	Savings (%)
0%	0.014	99.89	0.6	$0.0008	99.89
10%	11.91	99.50	3.5	$0.0021	99.73
20%	36.83	97.08	4.4	$0.0140	98.24
30%	48.75	71.47	18.0	$0.0977	87.73
40%	53.21	70.32	20.30	$0.1169	85.31
50%	63.71	54.66	30.16	$0.1265	84.10
60%	81.63	43.91	33.75	$0.1410	82.28
70%	90.59	29.46	37.77	$0.1470	81.53
80%	99.55	4.001	58.38	$0.1574	80.23
90%	99.86	0.858	59.80	$0.1708	78.54
100%	99.91	0.438	59.82	$0.1722	78.37
Average	61.52	46.85	29.80	$0.1042	86.91

TABLE II.
ANALYSIS OF BIDDING AT DIFFERENT CUMULATIVE PROBABILITY ON $c4.2xlarge$ $Linux$ SPOT INSTANCES OF US EAST (N. VIRGINIA)

Cumulative Probability	Successful Bids (%)	Interr-uption (%)	Average Lifetime (minutes)	Effective Price (per hr.)	Savings (%)
0%	0.009	100	0.005	$0.00	100
10%	06.91	72.13	04.49	$0.0066	98.34
20%	18.91	54.74	10.94	$0.0140	96.48
30%	27.87	35.36	11.84	$0.0479	87.96
40%	36.87	31.22	16.45	$0.0525	86.80
50%	45.87	21.22	26.15	$0.0626	84.27
60%	54.83	12.22	28.89	$0.0757	80.98
70%	72.67	11.89	36.30	$0.0752	81.11
80%	81.63	07.07	45.81	$0.0783	80.32
90%	81.63	02.98	50.53	$0.0781	80.37
100%	99.89	0.455	59.95	$0.0844	78.79
Average	48.93	22.84	25.33	$0.0577	85.50

Although the cumulative prediction probability of a certain percentage means the probability of the acceptance of the bid, that does not happen in most of the scenarios due to several reasons. There are two different cumulative probabilities- from the curve fitting and from the daily and weekly patterns. Among these two approaches, the curve fitting based prediction closely maintains the relationship between the bid acceptance and the cumulative probability. However, only the curve fitting is not reliable, as it is vulnerable to the market manipulation. As a result, a different acceptance distribution is observed compared to the cumulative probability density. The second column of both Table-I and Table-II present the percentage of successful bids. The percentage of successful bids always increases with the increase in the cumulative probability.

Less interruption is required for a high QoS. The third column of both Table-I and Table-II present percentage interruption for the corresponding spot instance. The percentage interruption is calculated as the ratio of the number of instances, interrupted within one hour of the acceptance and the number of accepted bids. According to our results, the percentage interruption is usually very high (close to 100%) while bidding at a very low price (0%-10% cumulative probability). Although the percentage interruption degrades the QoS, it is economically beneficial due to free partial hours.

The average availability value is calculated as the ratio of the sum of the lifetime of accepted bids in minutes (without the extra 2 minutes) and the total number of bids. As the result of the average duration can be significantly increased by a few longer-lasting bids. To overcome that limitation, when an instance survives for more than 60 minutes the lifetime is considered to be 60 minutes. When the average time is x minutes for a certain bid, the rough statistical progress from the bid is also x minutes. The fourth column of both Table-I and Table-II presents the average lifetime.

2) Cost Efficiency: The average price per hour in dollars is evaluated as the ratio of associated cost from non-terminated instances and the summation of lifetimes in hours. As a result, the average effective price becomes much lower when the percentage of interruption is higher. However, the result may vary from time to time. For example, while bidding at 0% cumulative probability, $c4.2xlarge$ instance received one bid and the bid is terminated after 10 minutes. Although bidding at such low cumulative probability results in a very poor QoS, the corresponding cost can be zero. Moreover, bidding at a lower value usually costs less but bidding at the median value can be more beneficial due to the partially used cost-free hours.

The last column of both Table-I and Table-II presents the percentage saving compared to the on-demand instances. The prices of $c4.2xlarge$ and $c4.4xlarge$ on-demand instances of US East (N. Virginia) location are $0.398 and $0.796 respectively. The proposed algorithm saves 86.91% and 85.50% respectively compared to the cost of the on-demand instances.

B. Performance with Distributed Bids

The information provided in Table-I and Table-II can directly help the type-2 user, bidding for a single instance. However, many customers of the Amazon SI are cloud brokers. Many of them have small tasks with varying deadlines. They also need a number of instances at a time. Some other users of scientific calculations can take multiple instances and perform the parallel execution but they want to complete the job within a few hours. These users are the type-1 users, according to our definition. They are prescribed to follow equation 5 for the calculation of their bid prices. The type-1 user gets the average performance when $n = 1$. The value of n varies over time based on the urgency of the task. When the task is not urgent, the value of n becomes close to 2. The proposed bidding system is designed to keep the value of n greater than 1 for most of the time, that helps all bidders by lowering the SI price. The left skewness is also observed in figure 4(a)-(c). The bid distributions in these figures are obtained as the percentile probability distribution according to equation (9); a part of the equation (5).

$$i^{th}\ Bid\ in\ Cumulative\ Probability = \{\frac{i}{N_{SI}+1}\}^n \quad (9)$$

N_{SI} is assumed as 1000 and the value of i is iterated from 1 to 1000. Bar charts are drawn from the bid distribution. As the user may approach with any arbitrary number of bids, bar charts, containing the bid count are normalized by the total number of bids. When the probability distribution function is a Gaussian one, $n > 1$ makes the bid distribution a negatively skewed Gaussian distribution, $n < 1$ results in a positively skewed Gaussian distribution, and $n = 1$ makes the bid distribution the same as the probability distribution.

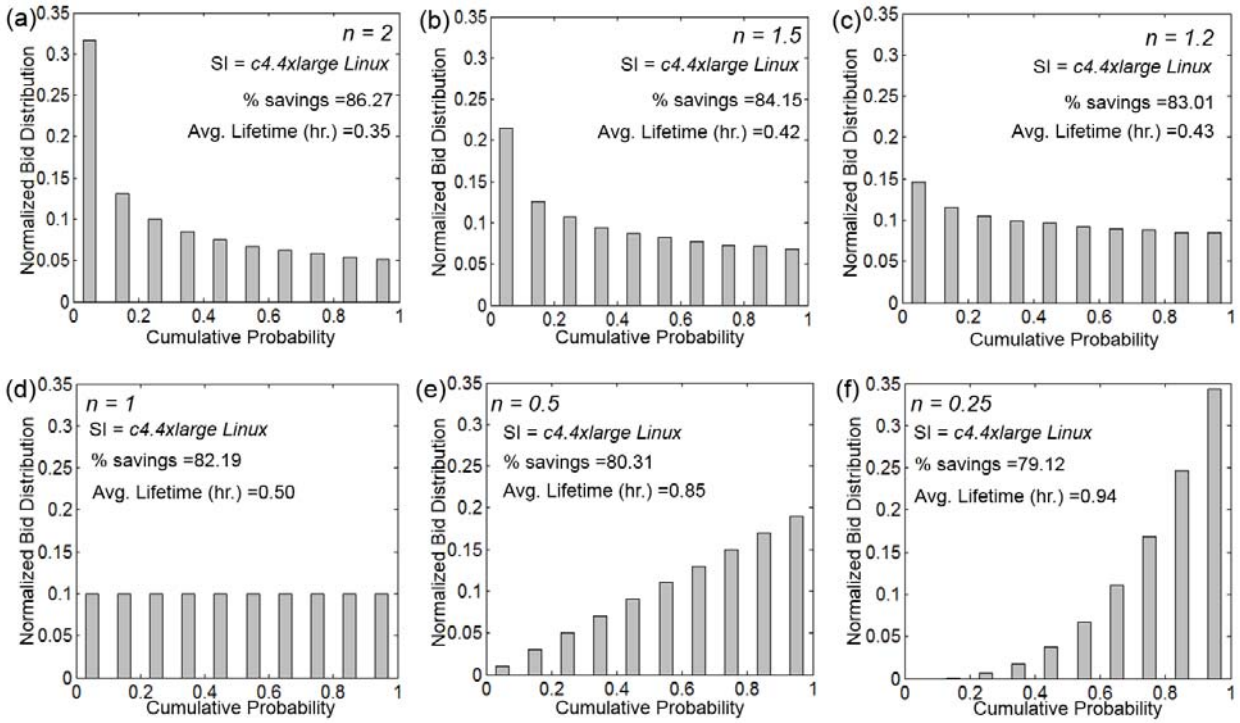

Fig. 4. Effective performance analysis for different bid distribution. Normalized bid distributions are obtained from equation 5. (a) $n = 2$, (b) $n = 1.5$, (c) $n = 1.2$, (d) $n = 1$, (e) $n = 0.5$, and (f) $n = 0.25$.

TABLE III.
MINIMUM BID PRICES REQUIRED FOR CERTAIN AVAILABILITY OF THE INSTANCE DURING A DAY (24 HOUR PERIOD)

Date	Required Availability of the Instance in Hours During One Day (24 hours) for $c4.4xlarge\ Linux$ Spot Instances of US East (N. Virginia)											
	1 hr.			2 hr.			4 hr.			8 hr.		
	Bid Price	Payment	Savings	Bid Price	Payment	Savings	Bid Price	Payment	Savings	Bid Price	Payment	Savings
11-03-'17	$0.1606	$0	100%	$0.1648	$0.1641	89.7%	$0.1687	$0.1641	94.9%	$0.1744	$0.3289	94.8%
12-03-'17	$0.1643	$0	100%	$0.1670	$0	100%	$0.1713	$0.1643	94.8%	$0.1797	$0.6812	89.3%
13-03-'17	$0.1693	$0.1677	78.9%	$0.1706	$0.1677	89.5%	$0.1769	$0.1677	94.7%	$0.1816	$0.3457	94.6%
14-03-'17	$0.1707	$0	100%	$0.1727	$0	100%	$0.1775	$0.1725	94.06%	$0.1819	$0.5278	86.5%
15-03-'17	$0.1547	$0	100%	$0.1563	$0.1551	90.3%	$0.1591	$0.1545	95.1%	$0.1662	$0.6372	90.0%
16-03-'17	$0.1479	$0	100%	$0.1485	$0.1475	90.7%	$0.1508	$0.4453	86.1%	$0.1562	$1.0556	83.4%
17-03-'17	$0.1582	$0	100%	$0.1634	$0	100%	$0.1687	$0	100%	$0.1740	$0.6159	90.3%
11-04-'17	$0.2726	$0	100%	$0.2748	$0	100%	$0.2767	$0.5477	82.8%	$0.2779	$1.3732	78.4%

TABLE IV.
PERFORMANCE EVALUATION OF JOBS WITH LONGER DEADLINE:
BIDDING AT A PRICE, PROVIDED CERTAIN COVERAGE ON THE PREVIOUS-DAY.

Date	Bid Price	Availability (Previous Day)	Availability (Current Day)	Payment	Savings	$\min(\frac{\text{Availability}_{\text{Current}}}{\text{Availability}_{\text{Previous}}}, 1) \times$ Savings
12-03-'17	$0.1687	4 hr. 00 min.	2 hr. 15 min.	$0	100%	56.25%
	$0.1744	8 hr. 00 min.	5 hr. 30 min.	$0.1692	96.1%	66.07%
13-03-'17	$0.1713	4 hr. 05 min.	2 hr. 35 min.	$0.169	91.8%	58.08%
	$0.1797	8 hr. 00 min.	6 hr. 30 min.	$0.169	96.7%	78.57%
14-03-'17	$0.1769	4 hr. 00 min.	4 hr. 10 min.	$0	100%	100%
	$0.1816	8 hr. 10 min.	7 hr. 50 min.	$0.3489	94.6%	09.74%
15-03-'17	$0.1775	4 hr. 10 min.	18 hr. 15 min.	$1.9559	86.5%	86.50%
	$0.1819	8 hr. 15 min.	20 hr. 40 min.	$3.0034	81.7%	81.70%
16-03-'17	$0.1591	4 hr. 00 min.	10 hr. 45 min.	$1.0499	87.7%	87.70%
	$0.1662	8 hr. 25 min.	15 hr. 25 min.	$1.5099	87.7%	87.70%
17-03-'17	$0.1508	4 hr. 05 min.	0 hr. 10 min.	$0	100%	04.08%
	$0.1562	8 hr. 10 min.	0 hr. 50 min.	$0	100%	10.20%
18-03-'17	$0.1508	4 hr. 05 min.	3 hr. 15 min.	$0	100%	79.59%
	$0.1562	8 hr. 00 min.	4 hr. 25 min.	$0.1676	95.3%	52.61%
Average	–	4 hr. 04 min.	5 hr. 54 min.	$0.4535	90.3%	90.33%
Performance	–	8 hr. 08 min.	8 hr. 44 min.	$0.7908	88.6%	88.56%

Figure 4 presents the bar chart of the normalized bid distribution for different values of n. According to graphs, the bidding density decreases with the increment the cumulative probability exponentially with the higher value of n and increases with the increment of cumulative probability exponentially with the lower value of n. Figure 4(a) presents the bar chart of the normalized bid distribution for $n = 2$. As most of the bids are at a lower cumulative probability many instances are terminated in the middle of an execution as a result, the savings compared to the on-demand is higher. However, the average lifetime of the bid is only 20.9 minutes. Figure 4(b) presents the bar chart of the normalized bid distribution for $n = 1.5$. Savings is slightly lower compared to $n = 2$, but the average lifetime of instances have increased to 25 minutes due to the low termination probability. Similarly, figure 4(c)-(f) are presenting bar charts of the normalized bid distribution for $n = 1.2$, $n = 1$, $n = 0.5$, and $n = 0.25$ respectively. The savings decreases and the average lifetime increases with the decrement of n. We do not suggest any particular value of n but the user needs to start bidding from a higher value of n and move towards lower values of n for the social welfare and proper bargaining.

C. Performance Evaluation of Jobs with Longer Deadline ($\geq$24 hr.)

Many scientific computations take several weeks in a low configuration server. Moreover, most research organizations do not need a high-end server for the whole year. They need time to plan the experiment and their bought servers become unutilized during the planning. Moreover, more researchers want to use the computing machine when the deadline for any project submission becomes close. These scientific computations, such as neural network training and atomistic simulations can allow several days of delay when the price is much cheaper. They require a certain amount of the instance occupancy for the completion. Therefore, the performance is evaluated in terms of the availability and the cost efficiency.

The proposed strategy for longer deadline jobs is to bid at a value that ensured certain availability on the previous day. Table III presents minimum bid prices required for certain availability of the instance during a day, corresponding payments, and savings. However, the price-curve varies from day to day. As the price curve of the next 24 hours is unknown, a successful example of previous day may fail to ensure the same availability. We suggest users for bidding at a price that covered two-time availability of the instance on the previous day compared to the required availability. Table IV presents the availability of the instance and corresponding savings while bidding with the 4hr. and 8hr. availability prices; obtained in the Table III.

1) Availability: The daily price-curve does not follow the exact pattern due to some random inauguration of jobs, weekly patterns, and yearly patterns. The fourth column of Table IV presents the availability of the instance while bidding at the 4-hour and 8-hour availability of the previous day. When the same pattern is repeated, the availability is also repeated. When the price becomes lower, the availability increases. When the price increases, the availability decreases. When a job is submitted considering the bid of the previous day, the job may fail to finish. Therefore, the user needs to re-bid after 24 hours by considering the price of recent 24 hours. Moreover, when the deadline is just several hours higher than the required time for the completion, the user needs to bid considering probability density of price, as mentioned in the first subsection.

2) Cost Efficiency: The fifth column of Table IV presents the payment and the sixth column presents the percentage savings. Through the process, we achieved 82% to 100% savings compared to the on-demand price($0.796). However, the savings are higher due to free partial hours. In some situations, the payment is $0 but the instance is available for less than one hour. To evaluate the performance, the savings is multiplied by the percentage availability and we achieve 4% to 100% performance. In one day the instance was available for more than the expected availability with frequent interruption. Therefore, no charging is experienced. We can expect that performance parameter effective when the job can be saved within the two-minute warning period.

D. Comparison between $c4.2xlarge$ and $c4.4xlarge$ Bidding

The functional difference between $c4.2xlarge$ and $c4.4xlarge$ instances are their capability. The $c4.4xlarge$ have 2 times the virtual central processing unit, memory and bandwidth compared to a $c4.2xlarge$ instance. According to the pricing curve analysis of different Amazon EC2 spot instances, the higher configuration servers have higher fluctuations in price. Therefore, the rate of interruption is higher for the $c4.4xlarge$ spot instances compared to $c4.2xlarge$; respectively 46.85% and 22.84% on average. However, maintaining a reasonable QoS with a high-performance non-reliable machine is more challenging, as a long time is required for the initialization of the job and saving the progress. As a result, many users of a high configuration machine agrees to increase the bid price when there is a slight price increase. It is good to increase the price when a significant amount of progress is not saved. The user needs to save the task immediately and needs to leave the instance during the further increase of the price. When users do not terminate, the bargaining does not exist.

V. Conclusion

The bargainer needs to know both the conditions of the market and his urgency. A proper bargaining can only bring the social welfare, resulting in an optimal profit to all users. In fact, there is no optimum thing in the market of bargaining. The user may refuse to pay a lower price when the task is not urgent. The user is willing to complete the task at a higher price when the task becomes urgent. Users need both short and long-term predictions for both the market and users' incoming tasks for a cost-efficient and reliable completion of their task. Our proposed algorithm proposes easily interpretable formulas for the bidding with the consideration of all of these issues.

From the providers' point of view, the provider can always modify their optimization function and a minimum acceptable bidding price to keep a certain profit margin. Moreover, when the spot market becomes larger due to its popularity, providers can serve on-demand and reserved instances more conveniently.

REFERENCES

[1] "Amazon ec2 spot instances," Jun. 2017. [Online]. Available: https://aws.amazon.com/ec2/spot/

[2] Q. Zhu and G. Agrawal, "Resource Provisioning with Budget Constraints for Adaptive Applications in Cloud Environments," *IEEE Transactions on Services Computing*, vol. 5, no. 4, pp. 497–511, 2012.

[3] A. N. Toosi, K. Van Mechelen, and R. Buyya, "An auction mechanism for a cloud spot market," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 11, no. 1, pp. 1–33, 2014.

[4] A. N. Toosi, K. Vanmechelen, F. Khodadadi, and R. Buyya, "An auction mechanism for cloud spot markets," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 11, no. 1, p. 2, 2016.

[5] L. Mashayekhy, M. M. Nejad, and D. Grosu, "Physical Machine Resource Management in Clouds: A Mechanism Design Approach," *IEEE Transactions on Cloud Computing*, vol. 3, no. 3, pp. 247–260, Jul 2015.

[6] A. N. Toosi, K. Vanmechelen, K. Ramamohanarao, and R. Buyya, "Revenue Maximization with Optimal Capacity Control in Infrastructure as a Service Cloud Markets," *IEEE Transactions on Cloud Computing*, vol. 3, no. 3, pp. 261–274, Jul. 2015.

[7] A. Marathe, R. Harris, D. K. Lowenthal, B. R. de Supinski, B. Rountree, and M. Schulz, "Exploiting redundancy and application scalability for cost-effective, time-constrained execution of hpc applications on amazon ec2," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2574–2588, 2016.

[8] S. Shastri and D. Irwin, "Towards index-based global trading in cloud spot markets," *HotCloud, June*, 2017.

[9] ——, "Cloud index tracking: Enabling predictable costs in cloud spot markets," *arXiv preprint arXiv:1809.03110*, 2018.

[10] D. Jung, S. Chin, K. Chung, H. Yu, and J. Gil, "An efficient checkpointing scheme using price history of spot instances in cloud computing environment," in *Network and Parallel Computing*, ser. Lecture Notes in Computer Science, E. Altman and W. Shi, Eds. Springer Berlin Heidelberg, 2011, vol. 6985, pp. 185–200.

[11] Y. Song, Z. Murtaza, and L. Kang-Won, "Optimal bidding in spot instance market," in *2012 Proceedings IEEE INFOCOM*. IEEE, Mar 2012, pp. 190–198.

[12] C. Qu, R. N. Calheiros, and R. Buyya, "A reliable and cost-efficient auto-scaling system for web applications using heterogeneous spot instances," *Journal of Network and Computer Applications*, vol. 65, pp. 167–180, 2016.

[13] S. Di and C. L. Wang, "Error-tolerant resource allocation and payment minimization for cloud system," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1097–1106, 2013.

[14] S. Subramanya, T. Guo, P. Sharma, D. Irwin, and P. Shenoy, "SpotOn: A Batch Computing Service for the Spot Market," in *Proceedings of the Sixth ACM Symposium on Cloud Computing - SoCC '15*. New York, New York, USA: ACM Press, Aug. 2015, pp. 329–341.

[15] I. Jangjaimon and N.-F. Tzeng, "Effective Cost Reduction for Elastic Clouds under Spot Instance Pricing Through Adaptive Checkpointing," *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 396–409, Feb. 2015.

[16] B. Javadi, R. K. Thulasiram, and R. Buyya, "Characterizing spot price dynamics in public cloud environments," *Future Generation Computer Systems*, vol. 29, no. 4, pp. 988–999, Jun. 2013.

[17] B. C. Tak, B. Urgaonkar, and A. Sivasubramaniam, "Cloudy with a Chance of Cost Savings," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1223–1233, Jun 2013.

[18] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of Resource Provisioning Cost in Cloud Computing," *IEEE Transactions on Services Computing*, vol. 5, no. 2, pp. 164–177, Apr. 2012.

[19] S. Tang, J. Yuan, and X.-Y. Li, "Towards optimal bidding strategy for amazon ec2 cloud spot instance," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. IEEE, 2012, pp. 91–98.

[20] C. Wang, B. Urgaonkar, A. Gupta, G. Kesidis, and Q. Liang, "Exploiting spot and burstable instances for improving the cost-efficacy of in-memory caches on the public cloud," in *Proceedings of the Twelfth European Conference on Computer Systems*. ACM, 2017, pp. 620–634.

[21] M. D. Blog, "Amazon ec2 spot request volatility hits $1000/hour," Jun. 2017. [Online]. Available: https://moz.com/devblog/amazon-ec2-spot-request-volatility-hits-1000hour/

[22] H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Neural network-based uncertainty quantification: A survey of methodologies and applications," *IEEE Access*, vol. 6, pp. 36 218–36 234, 2018.

[23] M. Alam, J. P. Parmigiani, and J. J. Kruzic, "An experimental assessment of methods to predict crack deflection at an interface," *Engineering Fracture Mechanics*, vol. 181, pp. 116–129, 2017.

[24] H. M. D. Kabir, "A study on the grain dependent current variation of polycrystalline organic transistors," Master's thesis, Hong Kong University of Science and Technology, 2016.

[25] H. D. Kabir, Z. Ahmed, L. Zhang, and M. Chan, "Coil-shaped electrodes to reduce the current variation of drop-casted otfts," *IEEE Electron Device Letters*, vol. 38, no. 5, pp. 645–648, 2017.

[26] Q. Sun, C. Wu, Z. Li, and S. Ren, "Colocation Demand Response: Joint Online Mechanisms for Individual Utility and Social Welfare Maximization," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3978–3992, Dec 2016.

[27] L. Zheng, C. Joe-Wong, C. W. Tan, M. Chiang, and X. Wang, "How to Bid the Cloud," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 5, pp. 71–84, Aug. 2015.

[28] L. Zhang, Z. Li, and C. Wu, "Dynamic resource provisioning in cloud computing: A randomized auction approach," *Proc. - IEEE INFOCOM*, pp. 433–441, 2014.

[29] D. Poola, K. Ramamohanarao, and R. Buyya, "Fault-tolerant workflow scheduling using spot instances on clouds," *Procedia Computer Science*, vol. 29, pp. 523–533, 2014.

[30] S. Alkharif, K. Lee, and H. Kim, "Time-series analysis for price prediction of opportunistic cloud computing resources," in *Proceedings of the 7th International Conference on Emerging Databases*. Springer, 2018, pp. 221–229.

[31] Z. Li, W. Tarneberg, M. Kihl, and A. Robertsson, "Using a predator-prey model to explain variations of cloud spot price," *arXiv preprint arXiv:1708.01397*, 2017.

[32] M. Lumpe, M. B. Chhetri, Q. B. Vo, and R. Kowalczyk, "On estimating minimum bids for amazon ec2 spot instances," in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE Press, 2017, pp. 391–400.

[33] D. Kumar, G. Baranwal, Z. Raza, and D. P. Vidyarthi, "A survey on spot pricing in cloud computing," *Journal of Network and Systems Management*, vol. 26, no. 4, pp. 809–856, 2018.

[34] A. S. Sabyasachi, H. M. D. Kabir, A. M. Abdelmoniem, and S. K. Mondal, "A resilient auction framework for deadline-aware jobs in cloud spot market," in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*. IEEE, 2017, pp. 247–249.

[35] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 1352–1372, 2015.

[36] R. Ghanem, D. Higdon, and H. Owhadi, *Handbook of uncertainty quantification*. Springer, 2017.

[37] H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Partial adversarial training for prediction interval," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–6.

[38] M. Malvoni, M. Fiore, G. Maggiotto, L. Mancarella, R. Quarta, V. Radice, P. Congedo, and M. De Giorgi, "Improvements in the predictions for the photovoltaic system performance of the mediterranean regions," *Energy Conversion and Management*, vol. 128, pp. 191–202, 2016.

[39] M. Malvoni, M. G. De Giorgi, and P. M. Congedo, "Forecasting of pv power generation using weather input data-preprocessing techniques," *Energy Procedia*, vol. 126, pp. 651–658, 2017.

[40] H. M. D. Kabir, A. Hosen, S. Nahavandi, and A. Khosravi, "Prediction Interval with Examples of Similar Pattern and Prediction Strength," in *The 30th Annual IEEE Canadian Conference On Electrical and Computer Engineering, CCECE 2017*. IEEE, 2017, pp. 1–4.