Performance and Behavior Characterization of Amazon EC2 Spot Instances

Thanh-Phuong Pham
University of Innsbruck
Innsbruck, Austria
phuong@dps.uibk.ac.at

Sasko Ristov
University of Innsbruck
Innsbruck, Austria
sashko@dps.uibk.ac.at

Thomas Fahringer
University of Innsbruck
Innsbruck, Austria
tf@dps.uibk.ac.at

***Abstract*—Amazon EC2's spot instances (SIs) represent a competitive Cloud resource in terms of price compared to reliable and fixed price options. The drawback, however, is that SIs may not always be available and they can be revoked at any given time. In this paper, we describe a comprehensive experimental evaluation for EC2 SIs to characterize their performance and behavior in three different regions each of which in a different continent. We describe the life cycle of SIs with the most important phases of an SI, introduce the most relevant events that can prevent a user from obtaining SIs, and draw important conclusions that can be exploited by the research community to effectively use the spot market. Our results reveal the fulfillment rate of requests for SIs, waiting time until requested SIs become fulfilled, details about the interruption rate of SIs, and how long SIs run before being interrupted. Our study also indicates that the SI interruption rate influences the fulfillment rate, SIs are highly reliable in the first 30 minutes after deployment, existing biding strategies are likely to fail for SIs with medium interruption rate, and SIs can be reclaimed by EC2 anytime regardless of an SI's bid price or workload.**

***Index Terms*—EC2; interruption; reliability; spot instances;**

I. Introduction

Amazon Elastic Compute Cloud (EC2) offers elastic resources upon user demands which can be provisioned immediately even if the system is overloaded [1]. However, when its computing resources are under-provisioned, EC2 can rent spare unused resources as spot instances (SIs) [2] at a much cheaper price than for On-Demand instances based on an auction bidding mechanism. The trade-off for cheaper prices is the possibility of EC2 to reclaim some or all SIs in case the current spot price is greater than the user's bid price or if EC2 lacks resources for On-Demand or Reserved instances. EC2 publishes a Spot Pricing History for all offered SIs for all availability zones in each region (data center) of the world for the last three months in order for users to become aware of recent price changes for SIs. This data has been used to model the future price behavior of SIs and to develop bidding strategies that can mitigate the *interruptions* of SIs [3], [4].

EC2 also publishes the level of *interruption rate* (low, medium or high) for each SI type in each region, without providing information on the availability zone for these regions. This data can be useful to derive a bidding strategy by selecting SIs with low interruption rate. However, our analysis shows that such strategies lack important information that can be fundamental to effectively use the spot market of public Cloud infrastructures.

This paper focuses on the research question: *How much can a user rely on EC2 SIs*? In order to find an answer to this question, we conducted a series of experiments by submitting a total of 3840 requests for SIs in three EC2 regions (each of which in a different continent). We explored 20 SI types, each with varying workloads and bid prices in order to elaborate whether EC2 prefers to interrupt SIs with specific workloads or bid prices. We investigate important parameters and discover insights which have not appeared in literature so far. For example, we determine the distribution of fulfillment rate for requested SIs, classify the waiting time for fulfilled SIs, analyze the reasons for unfulfilled SIs and examine the interruption rate and running time of SIs before an interruption occurs.

There are a limited number of works on modeling the behavior of SIs in a simulated environment [5], [6], [7], [8], [9], [10], whereas our evaluation is based on real EC2 cloud experiments. Previous work simplifies the behavior of SIs at the price of an inaccurate modeling for the spot market. We introduce a taxonomy for a more accurate spot instance lifecycle with the most important events and phases. For example, [11], [12], [13] defined bidding strategies that mitigate the risk of interruptions, but only by keeping the bid price above the spot price of SIs without considering the interruptions due to low capacity. Other researchers described a low cost dynamic scheduling algorithm for SIs without considering the waiting time for obtaining SIs [14].

The main observations based on our experimental study for EC2 SIs are:

- SI types with higher level of interruption rate (reported by EC2) also tend to have shorter runtimes before interrupts, smaller fulfillment rate and longer times until fulfillment. We thus conclude that the reliability (described by its interruption rate) of an SI type influences also how much and how often SIs are available for users.
- If a request for a specific SI type is not fulfilled within four seconds, it is very unlikely that this request will be fulfilled within one minute. For such scenarios it is advised to switch to another SI type, which could be fulfilled faster instead of waiting for the original SI type.

- SIs are notably reliable in the first 20-30 minutes after deployment even those with medium level of interruption rate. For example, only 10% of SIs with low and 20% with medium level of interruption rate were interrupted in the first 30 minutes.
- Bidding strategies are useful and can deal mostly with SI types with low level of interruption rate. However, they are likely to fail for SI types with medium level of interruption rate, which are mostly interrupted due to lack of capacity.
- EC2 can also reclaim capacity regardless of an SI's workload or its bid price.
- 78% of requests for SI types with low level of interruption rate and 40% with medium level of interruption rate behaved largely identical as On-Demand resources for the first four hours of runtime, respectively. Others requests for SIs were either unfulfilled, or interrupted by EC2.

The rest of the paper is organized in several sections. Section II introduces the concept and main terminology of SIs. In Section III, we explain our evaluation method for EC2 SIs, while in Section IV we describe experiments and the resulting performance and behavior of SIs. Specific insights of the evaluation are discussed in Section V, followed by Section VI, which compares our approach with related work. Finally, Section VII concludes the paper and outlines our plans for future work.

II. Background

A. EC2 Spot Instances

EC2 offers spare unused (sometimes called volatile) computing resources in form of SIs at a discount of up to 90% compared to their fixed price reliable On-Demand instances [2]. SIs, however, can be interrupted by EC2 at any time and are thus less reliable.

SIs are offered in two different forms: *fleet* or *block*. When SIs are rented in a form of a fleet, users only need to specify the amount of resources they need. When SIs are rented as a block, a user can run an SI for a predefined hourly-based time period for up to six hours. In this case, the SIs are not interrupted, but savings are lower commonly within 30-50% compared to On-Demand prices.

B. SI terminology

The terminology about SIs is described in Table I, which will be used throughout the paper. Note that time events t_i are used in this table which refer to time events shown in Fig. 1.

In order to get access to an SI, a user must submit a spot request in which they specify the bid price, which should be larger or equal to the current spot price. In contrast to On-Demand instances with fixed prices for the entire usage period, the spot price of an SI may change at intervals of five minutes. In case the spot price increases above the bid price, EC2 will interrupt the SI and reclaims it from the user. A lack of computing resources to satisfy requests for On-Demand or Reserved instances can be another reason for EC2 to interrupt a running SI.

TABLE I
Terminology for EC2 spot instances

Definition	Description
spot request	user submits a request for an SI
bid price	bid price set by the user in a spot request
spot price	the current price of an SI in the spot market
status message	the current status of a spot request
fulfilled spot request	spot request is approved by EC2 and an SI starts to deploy
waiting time	time between submitting a spot request and fulfilled spot request: $t_3 - t_1$
deployment time	time required to deploy an SI once the spot request is fulfilled: $t_4 - t_3$
running time	time between an SI that is deployed until it is terminated: $t_5 - t_4$ or $t_8 - t_4$
warning time	time between SI interruption until termination: $t_6 - t_5$

C. SI Life Cycle

Fig. 1 illustrates the life cycle of an EC2 SI, starting from the spot request until the SI is terminated. After submitting a spot request, a user has to wait until the request is fulfilled. EC2 may respond to a request with so called status messages indicating that there is currently no capacity or the bid price is below the spot price. A request can be canceled until it becomes fulfilled. Once the spot request is fulfilled, the SI has to be deployed and afterwards the SI can be used by the user until

- the SI is terminated by the user, or
- a notification is sent by EC2 that the SI will be interrupted either due to a spot price greater than the bid price or EC2 lacks capacity for On-Demand or Reserved instances.

Shortly after the interrupt, the SI will be terminated by EC2. EC2 charges the user of an SI only for the running time but not for the deployment time.

A spot request can finish in one of these three final states:

- *unfulfilled spot request* - spot request not fulfilled due to lack of resources or low bid price,
- *fulfilled spot request with interruption* - deployed SI, but later interrupted due to low bid price or lack of capacity,
- *fulfilled spot request without interruption* - deployed SI and terminated by the user once the SI is no longer needed.

D. SI status messages

Table II describes the most important status messages invoked by EC2 for every specific event associated with an SI. If EC2 replies to a spot request with *no-capacity*, a user can cancel the request anytime before a deadline (indicated as part of the spot request) is reached unless EC2 offers the requested resources. If no resources are offered by EC2 until the end of this time period, then the request will be automatically canceled. However, if the user submits a spot request with a bid price lower than the spot price, EC2 will reply with *low-bid-price* and the status of the spot request remains open. When the bid price is greater or equal to the spot price, and EC2 has available resources, then the *fulfilled request* status message

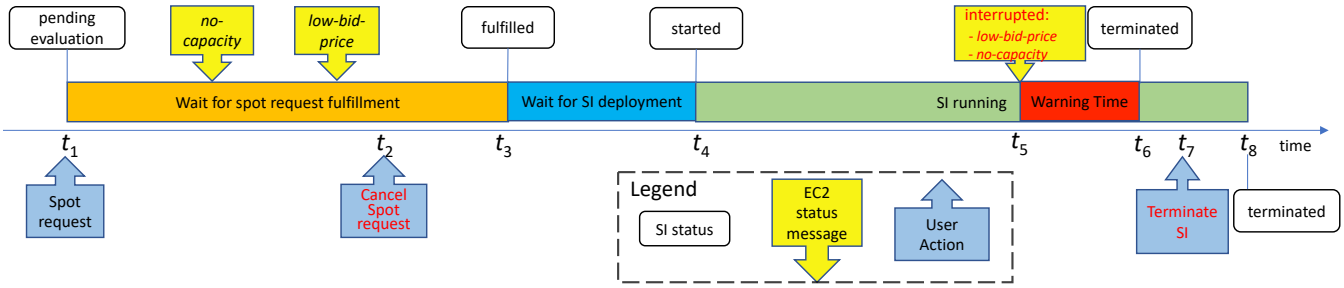

Fig. 1. EC2 spot instance life cycle with important time events, status messages and user actions

TABLE II
SI STATUS MESSAGES BY EC2

Status Message	Description
no-capacity	no spot capacity available
low-bid-price	bid price of spot request lower than current spot price
fulfilled request	specifications for a spot request are fulfilled and the SI is ready for deployment
interrupted	SI is interrupted by EC2
terminated	SI is terminated (by EC2 or a user)

is set by EC2 and the SI can be deployed. Thereafter, if the spot price rises above the bid price, the status of spot request changes to *interrupted*. The *no-capacity* and *low-bid-price* status messages before a spot request is fulfilled will put this request on hold and there is still a chance for this request to be fulfilled unless the user cancels the spot request explicitly. An *interrupted* status message invoked for a specific instance is irreversible and leads to a termination of this instance after approximately two minutes.

III. EVALUATION METHOD

This section presents our method for evaluating the behavior and performance of EC SIs. We determined all relevant events associated with SIs (for various SI types and different EC2 regions) which can differ from On-Demand instances. We measured the frequency of these events described in Section II-C as well as the time duration between these events. We also examined whether the bid price or workload of a fulfilled SI influences the likelihood for being interrupted.

A. Experimental setup

We have conducted our experiments with a large set of instance types on three different EC2 regions: North Virginia $\mathcal{V}$ (availability zone us-east-1b), Frankfurt $\mathcal{F}$ (availability zone eu-central-1b) and São Paulo $\mathcal{SP}$ (availability zone sa-east-1c) as presented in Table III. All data shown in this table has been reported by the EC2 web site[1] and APIs on January 22, 2018. The focus for these experiments was primarily on general purpose (type t or m), compute optimized (type c) and storage optimized (type d) instance types which are predominantly used for scientific computing [1], [14], [15], [16].

[1]https://aws.amazon.com/ec2/instance-types/

TABLE III
EVALUATED SI TYPES WITH INTERRUPTION RATE (IR) AND AVERAGE PRICE DISCOUNT (PD) COMPARED TO ON-DEMAND IN EACH REGION $\mathcal{F}$, $\mathcal{SP}$ AND $\mathcal{V}$ AS SPECIFIED BY EC2

SI type (CPUs, RAM(GB), Network(Gb/s))	$\mathcal{F}$		$\mathcal{SP}$		$\mathcal{V}$	
	IR	PD	IR	PD	IR	PD
c3.l (2, 3.75, M)			*l*	78%	*l*	73%
c3.xl (4, 7.5, M)					*l*	73%
c3.2xl (8, 15, H)					*l*	69%
c3.4xl (16, 30, H)	*l*	65%	*l*	*	*l*	68%
c3.8xl (32, 60, 10 Gb)	*l*	62%	*l*	*	*l*	73%
c5.l,(2, 4, $\leq$10 Gb)					*l*	64%
c5.4xl (16, 32, $\leq$10 Gb)					*l*	57%
m4.2xl (8, 32, H)	*l*	74%	*m*	57%		
m4.4xl (16, 64, 2 Gb)	*l*	74%	*m*	56%		
m4.10xl (40, 160, 4 Gb)	*l*	66%	*m*	56%		
m4.16xl (64, 256, 25 Gb)	*l*	62%	*m*	56%		
d2.xl,(4, 30.5, M)					*l*	70%
d2.2xl,(8, 61, H)					*l*	75%
d2.4xl,(16, 122, H)					*l*	70%
cc2.8xl (32, 60, 10 Gb)					*l*	84%
t2.s,(1, 2, L to M)					*l*	70%
t2.med,(2, 4, L to M)					*l*	70%
t2.l,(2, 8, L to M)					*l*	70%
m5.l,(2, 8, $\leq$10 Gb)					*l*	68%
m5.xl,(4, 16, $\leq$10 Gb)					*l*	68%

We evaluated instance types with various *resources* (1-64 CPUs, 2-256GB memory, network speed from 100Mbps up to 25Gbps), *level of interruption rate* IR (low l or medium m) and *price discount* PD ($56 - 84\%$) compared to On-Demand equivalent instances. For example, EC2 reports that m4.2xlarge and m4.4xlarge have low level of interruption rate in all three regions, whereas m4.10xlarge and m4.16xlarge have a medium level of interruption rate. Another interesting observation is that there are cases (for c3.4xlarge and c3.8xlarge, marked with "*" in Table III) where EC2 does not offer a corresponding SI type for an On-Demand instance, although the price discount rate is specified for such SI types.

B. Pricing and Workloads

When EC2 lacks resources for On-Demand or Reserved instances, then it may occur that deployed SIs are interrupted. However, there is little information about their interruption policy. Therefore, we examine whether the bid price or workload impacts the interruption behavior of SIs. For example,

does EC2 prefer to interrupt SIs with lower bidding price or maybe those with higher workload.

1) Bid prices: In order to evaluate the correlation between biding strategies and interruption rate of SIs due to the lower bid price, we used four different bid prices p_b, the spot price and three bid prices for $\alpha = 0.25, 0.5$ and 0.75 of the difference between the on-demand price p_o and spot price p_s, as presented in (1).

$$p_b = p_s + \alpha \cdot (p_o - p_s) \tag{1}$$

2) Workloads: We also explored whether different workloads influence the interruption rate of an SI. For this purpose, we have developed four benchmarks applications with low and high CPU and memory loads that we run in each deployed SI.

Low: low computational and memory load implemented as a matrix-matrix multiplication of size 1000. The *Low* benchmark is repeatedly executed with a sleep command of 20 seconds between each execution, which provides a negligible additional CPU and memory utilization.

CPU: compute bound workload that uses the full computational power of an SI's CPU. We run continuously an OpenMP version of a dense matrix-matrix multiplication to exploit all CPUs of the SI. This benchmark results in 100% CPU and only up to 2.4% memory usage on the SI.

MEM: memory bound workload that allocates the full RAM memory of the SI, releases it after one minute and then reallocates it again. This benchmark results in 100% memory and a negligible CPU utilization.

CPU+MEM: compute and memory bound workload runs both *CPU* and *MEM* benchmarks, thus resulting in 100% CPU and memory utilization.

C. Experimental Setup

We created a total of 3840 spot requests organized in experiments each of which invokes 16 requests to the same SI type. Those 16 requests have been sub-divided into 4 different workloads and bid prices as described in the previous Section Section III-B. We run six experiments concurrently with a total of 96 spot requests due to the spot limit of 100 set by EC2. All experiments have been conducted only during working days of January 2018.

We submitted each spot request with a duration of 4 hours and canceled it if it was not fulfilled within that time frame. For fulfilled spot requests, we run specific benchmarks as described in Section III-B for up to 4 hours unless EC2 terminates the SI ealier. We have chosen a maximum runtime of 4 hours for each deployed SI as many scientific tasks can be finished within that time frame [15] and also in order to keep the total costs (EURO 3.230,-) of our experiments under our available budget, which was EURO 3500,-. We instrumented and monitored all events (SI status changes, status messages, user actions, time events) as shown in Fig. 1 for all spot requests and deployed SIs. The time events are used to measure the various phases for each spot request (wait for spot request to be fulfilled, wait for SI deployment, running time, etc.).

TABLE IV
SPOT REQUEST FULFILLMENT RATE

	$\mathcal{F}$	$\mathcal{SP}$	$\mathcal{V}$
unfulfilled	4 (0.8%)	360 (25.0%)	4 (0.2%)
fulfilled	476 (99.2%)	1080 (75.0%)	1916 (99.8%)
total	480	1440	1920
experiments (=total/16)	30	90	120

We used EC2 APIs to query the status of each active spot request every 10 seconds, starting from submitting the spot request at time t_1, until canceling the spot request at time t_2, or termination of SIs at times t_6 or t_8. After a spot request is fulfilled and the public IP address of an SI can be obtained, we store that time as the SI deployment time and start an appropriate workload benchmark on an otherwise unloaded SI (only the benchmarks runs on it). We then check every 10 seconds the CPU and memory utilization to ensure that the benchmarks are still running.

IV. EVALUATION

This section describes experiments and characterizes the behavior and performance of SIs.

A. Spot requests fulfillment

Table IV illustrates the spot request fulfillment rate for three EC2 regions. More than 99% of all spot requests have been fulfilled for $\mathcal{F}$ and $\mathcal{V}$, and 75% have been fulfilled for $\mathcal{SP}$.

We examined the fulfillment rate for every experiment each of which invoked 16 spot requests to the same SI type. Fig. 2 shows the number of fulfilled spot requests for region $\mathcal{SP}$. The other two regions $\mathcal{F}$ and $\mathcal{V}$ have only one experiment each with 12 fulfilled and 4 unfulfilled spot requests. All other experiments, 29 in $\mathcal{F}$ and 119 in $\mathcal{V}$, were fulfilled (all 16 spot requests for every experiment). For region $\mathcal{SP}$ we observed that only 29 out of 90 experiments had at least one unfulfilled spot request, and 12 experiments had zero fulfilled spot requests. For these experiments we could not determine any correlation between the bid price and unfulfilled spot requests. However, we conclude that resource capacity has higher impact on the fulfillment rate since for the m4.16xlarge instance (most powerful instance which we examined for our experiments), EC2 fulfilled only 25 out of 176 spot requests. For all experiments with the m4.16xlarge instance type, there was not a single experiment with all requests for this instance type fulfilled.

B. Classifying Waiting Time of fulfilled spot requests

Since some of the spot requests were not fulfilled immediately, we analyzed the waiting time until the spot request was fulfilled, which is given by the time interval $t_3 - t_1$ in Fig. 1. It should be noted that On-Demand instances are provisioned immediately.

Table V classifies the number of spot requests across three different regions with respect to waiting times. The entry 431 (90.5%) for region $\mathcal{F}$ means that 90.5% (total number 431)

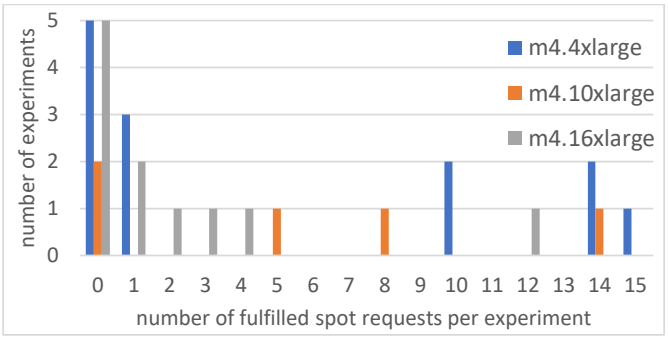

Fig. 2. Fulfillment rate of experiments for region $\mathcal{SP}$ with up to 15 (out of 16) fulfilled spot requests

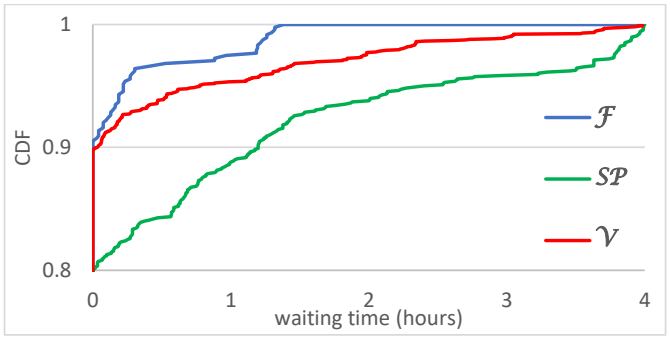

Fig. 3. Cumulative distribution for the waiting time ($t_3 - t_1$) of fulfilled spot requests in 3 EC2 regions

TABLE V
WAITING TIME FOR FULFILLED SPOT REQUESTS

waiting time wt [secs]	$\mathcal{F}$	$\mathcal{SP}$	$\mathcal{V}$
wt ≤ 4	431 (90.5%)	865 (80.1%)	1715 (89.5%)
$4 <$ wt ≤ 60	0 (0.0%)	1 (0.001%)	2 (0.1%)
$60 <$ wt ≤ 14.400	45 (9.5%)	214 (19.8%)	199 (10.4%)

TABLE VI
INTERRUPTION RATE OF SIS

fulfilled spot requests	$\mathcal{F}$	$\mathcal{SP}$	$\mathcal{V}$
without interrupt	415 (86.5%)	713 (66%)	1680 (87.5%)
with interrupt	61 (12.7%)	367 (34%)	236 (12.5%)

TABLE VII
REASONS FOR INTERRUPTIONS OF SIS

Reason for interruption	$\mathcal{F}$	$\mathcal{SP}$	$\mathcal{V}$
low-bid-price	16 (26.2%)	8 (2.2%)	70 (29.7%)
no-capacity	45 (73.8%)	359 (97.8%)	166 (70.3%)

of all 476 spot requests have been fulfilled within 4 seconds. $\mathcal{SP}$ had the smallest rate of *immediately* (less than 4 seconds) fulfilled spot requests which was 80.1%, while the others had higher rates of 89.5% and 90.5% in $\mathcal{V}$ and $\mathcal{F}$, respectively. The longest waiting time (larger than 60 seconds) has been exposed by $9.5 - 19.8\%$ spot requests across the three regions where $\mathcal{SP}$ represents the largest number of spot requests with long waiting times. Medium waiting times (between 4 and 60 seconds) have been rarely observed. Based on our experiments, we can conclude that at least 80% of the SIs can be fulfilled within 4 seconds. In most other cases we have to wait for more than 60 seconds. This information can be useful for resource provisioning and scheduling algorithms for Clouds [17], [18].

Fig. 3 illustrates the cumulative distribution function (CDF) for waiting time for the three evaluated EC2 regions. Although $\mathcal{F}$ and $\mathcal{V}$ have similar waiting time behavior for fulfilled spot requests according to Table V, $\mathcal{F}$ fulfilled 100% spot requests in less than 1.38 h. Whereas $\mathcal{V}$ required 3.96 h to fulfill all spot requests. Region $\mathcal{SP}$ had the highest rate of *long* waiting times (more than 60 seconds). These spot requests also took longer to be fulfilled compared to $\mathcal{F}$ and $\mathcal{V}$.

C. Characterization of SI interruptions

Once an SI is provisioned and deployed, it can be interrupted by EC2 after *no-capacity* or *low-bid-price* status messages. Table VI presents the interruption rates for the three evaluated EC2 regions. EC2 regularly updates and publishes the level of interruption rate for all regions. During our experiments EC2 reported for $\mathcal{F}$ and $\mathcal{V}$ a low level of interruption rate whereas for $\mathcal{SP}$ a medium level of interruption rate was stated. Based on our experiments, we observed that $\mathcal{F}$ and $\mathcal{V}$ had similar interruption rates of approx. 12.5%, while $\mathcal{SP}$ resulted in 34% interruptions which correlates with the published low level of interruption rates for evaluated SIs in $\mathcal{F}$ and $\mathcal{V}$, and medium level of interruption rates for evaluated SIs in $\mathcal{SP}$, as announced by EC2.

Table VII analyzes the reasons for SI interruptions. We observe that EC2 interrupts SIs mostly because it lacks capacity for On-Demand or Reserved instances. Interrupting SIs due to low bid prices specified by users in spot requests occurs less often (2.2 % - 29.7 %).

For regions $\mathcal{F}$ and $\mathcal{V}$ only a few instance types caused interrupts by EC2, whereas $\mathcal{SP}$ resulted in interrupts for at least one SI for every evaluated instance type. In the following subsections we explore whether a correlation of the deployed SIs exists between the interruption rate and either the bidding price or the workload. For this analysis we consider only the interruptions caused by status message *no-capacity*, since EC2 will interrupt all SIs based on *low-bid-price* regardless of their workload.

1) Interruption rate and bid price correlation: The four α-columns in Table VIII tabulate the interruption rate for each SI type for different bid prices in the three EC2 regions. Although one would expect that EC2 will first interrupt SIs with lower bid prices, our experiments did not uncover any considerable correlation between bid price and interruption rate. The interruptions appear to be close to uniformly distributed for all values of the α (bid price) parameter.

Two scenarios are representative for EC2's behavior to select an SI for interruption. For example, for the instance type c3.8xlarge in $\mathcal{F}$, EC2 selected SIs with the highest bid price for being interrupted in 33% of the total number of interruptions. Lower bid price spot requests ($\alpha = 0.25$) have

TABLE VIII
DISTRIBUTION OF INTERRUPTIONS FOR DIFFERENT α PRICE RANGES AND WORKLOADS

Reg.	SI Type	$\alpha(\%)$				workload			
		0	25	50	75	L	C	M	CM
$\mathcal{F}$	c3.4xl	50	0	50	0	50	0	0	50
$\mathcal{F}$	c3.8xl	25	17	25	33	12	20	20	48
$\mathcal{F}$	m4.10xl	17	33	25	25	25	33	25	17
$\mathcal{SP}$	c3.4xl	33	33	11	33	30	10	20	40
$\mathcal{SP}$	c3.8xl	26	22	24	28	28	30	18	24
$\mathcal{SP}$	m4.2xl	21	29	21	29	32	26	21	21
$\mathcal{SP}$	m4.4xl	33	21	23	23	24	29	27	20
$\mathcal{SP}$	m4.10xl	31	23	22	24	25	26	25	24
$\mathcal{V}$	c5.xl	0	33	67	0	0	0	33	67
$\mathcal{V}$	c5.2xl	25	25	50	0	25	25	50	0
$\mathcal{V}$	c5.4xl	12	25	38	25	25	25	31	19
$\mathcal{V}$	cc2.8xl	22	27	26	25	25	24	27	24

been interrupted for 17% of the total number of interruptions. Similarly, for c5.xlarge in $\mathcal{V}$, only SIs with bid price $\alpha = 0.25$ and $\alpha = 0.5$ were interrupted. Overall, we can conclude that when EC2 runs out of capacity, it will interrupt SIs regardless of the associated bid price.

2) Interruption rate and SI workload correlation: The four workload columns in Table VIII show the interruption rates for SI types with various workloads in the three EC2 regions. Although one would expect that EC2 will interrupt the instances with a higher workload, nevertheless, we could not find any correlation between the interruption rate and the workload. For example, for the instance type c5.2xlarge in $\mathcal{V}$, we have not detected any interruption for the CPU+MEM workload. Similarly, for c3.4xlarge in $\mathcal{F}$, SIs with low workload were interrupted, although there were instances with higher loads (CPU+MEM).

D. Running time of deployed SI with interruption

We continue our evaluation by analyzing how long a deployed SI will run before being interrupted. The running time of an interrupted SI is presented in Fig. 1 as the time period from t_4 to t_6. Fig. 4 illustrates the CDF for the running time of deployed SIs that were interrupted, for the three evaluated regions. Higher curves mean that more interruptions happen earlier, while lower curves reflect that SIs in that region are more reliable. Among those regions that we examined in our study, we observed that $\mathcal{F}$ is the most reliable region. Region $\mathcal{V}$ is less reliable but still better than $\mathcal{SP}$.

We can draw several important conclusions from the CDF for the running time of SIs. For example, half of all interrupted SIs run at least 3 h in $\mathcal{F}$, less than 2 h in $\mathcal{V}$, and less than 1.5 h in $\mathcal{SP}$. It is also interesting to examine how long SIs run until interruption. For example, 93.5% of SIs run at least 30 minutes in $\mathcal{F}$, which means that only 6.5% are interrupted within the first half hour. Similar observation can be made for $\mathcal{V}$, where 8.66% are interrupted within 30 minutes or only 4.8% within the first 20 minutes. $\mathcal{SP}$ was less reliable with a total of 8.58% of SIs that were interrupted within 20 minutes.

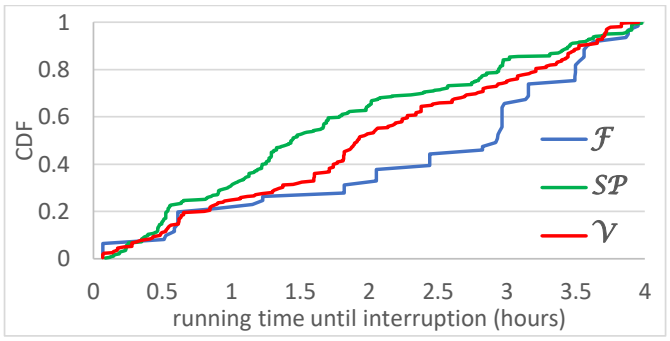

Fig. 4. Cumulative distribution for the running time ($t_6 - t_4$) of deployed SIs with interruption in 3 EC2 regions

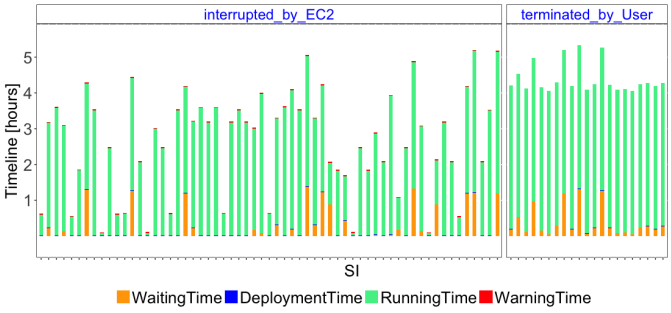

Fig. 5. SIs interrupted by EC2 and terminated by the user (without interruption by EC2 after four hours) in $\mathcal{F}$

E. Behavior of fulfilled spot requests

Fig. 5 illustrates the behavior of fulfilled spot requests in region $\mathcal{F}$, as it had the lowest number of fulfilled spot requests. For every SI we show the waiting, deployment, running and warning times which are described in Table I. The left part shows all 61 SIs interrupted by EC2. The right part displays only those 21 SIs with waiting time greater than 60 seconds, which were terminated by the user four hours after the deployment. The remaining SIs are not shown since their waiting time is less than 4 seconds and all of them have identical running time of 4 hours. By analyzing the interrupted SIs in Fig. 5, we do not observe any correlation between waiting and running times for SIs. We thus cannot conclude for spot requests that were fulfilled immediately, that they will be often or rarely interrupted after the deployment of SIs. There is also no evidence based on our experiments for a correlation for waiting times between the interrupted SIs and those terminated by the user.

Regarding the deployment time, we can report a mean value of $38secs$ ($\sigma = 5.5secs$) for all SIs shown in Fig. 5, which is similar to $42secs$ ($\sigma = 4.6secs$) for On-Demand instances [19]. Although in principle it may be possible that EC2 interrupts an SI during deployment time, we never encountered such a scenario for any of our experiments. Finally, all terminations of interrupted SIs occurred two minutes after the $interruption$ status message was invoked by EC2. This means

that the warning time as shown in Fig. 1 was two minutes for all interrupted SIs in all three regions of our study.

V. Discussion

In this section, we discuss the main insights from our evaluation, which can be used by the research community to effectively use the spot market. We examine possible correlations between information that is publicly available at the EC2 web site and various events and timings that we have observed for SIs.

A. Characterization of spot requests before fulfillment

We analyzed the status messages (*no-capacity*, *low-bid-price* or *fulfilled*) for all spot requests starting from t_1 until t_3 (see Fig.1). For some spot requests EC2 invoked status messages *no-capacity* or *low-bid-price* yet at a later stage they still became *fulfilled*. EC2 does not terminate a spot request based on these status messages. Only the user can cancel such requests.

Immediately before we request for an SI, we first determine the current spot price by using the EC2 API. Our bid price was at least as large as the current spot price, thus we expected that all status messages for our spot requests will either be *no-capacity* or *fulfilled*. This was true for all but 3 experiments with a bid price equal to the latest spot price. 12 spot requests of these 3 experiments returned a status message *low-bid-price*. We also observed (although rarely) a severe price increase of 42% (from $0.0602\$$ to $0.0856\$$ per hour) between requesting the current spot price and submitting a request for 16 SIs for one of these experiments. For the other two experiments the price slightly increased by 0.3% and 0.8%, respectively which also resulted in a status message *low-bid-price*. For most experiments spot prices went up after fulfillment of spot requests.

B. SI versus On-Demand resources

Table V tabulates a high rate of immediately fulfilled spot requests without waiting time from 80 % for $\mathcal{SP}$ up to 90 % for $\mathcal{V}$ and $\mathcal{F}$. For these requests, EC2 directly replied with status message *fulfilled*. If we cross correlate these results with Table IV, we can determine that almost 90% of spot requests are immediately fulfilled in $\mathcal{V}$ and $\mathcal{F}$, but only 60% for $\mathcal{SP}$.

If we combine the interruption rate from Table VI with immediately fulfilled spot requests, we can infer that more than 78% of all spot requests were immediately fulfilled and later not interrupted for at least four hours in $\mathcal{V}$ and $\mathcal{F}$. On the other side, in $\mathcal{SP}$ only 40% of submitted spot requests were immediately fulfilled and later not interrupted for at least four hours. These SIs thus had a similar behavior as On-Demand resources for a period of four hours.

We also analyzed the cost-reliability trade-off between SIs and equivalent On-Demand instances. Namely, the average discount for SIs in $\mathcal{V}$ and $\mathcal{F}$ was 67% to 70%. Considering the ratio of SIs that behaved similar as On-Demand instances, we can conclude that by reducing the reliability by 22%, one can gain approximately 70% cost reduction.

C. Level of interruption rate cross-correlation

The level of interruption rate for SIs as published by EC2 for the three regions (low in $\mathcal{F}$ and $\mathcal{V}$ and medium in $\mathcal{SP}$) also correlates with the total number of fulfilled spot requests (including those with waiting time larger than zero until fulfillment). 25% of submitted spot requests in $\mathcal{SP}$ were unfulfilled (Table IV), which is much higher than for $\mathcal{F}$ and $\mathcal{V}$ (less than 1%).

Furthermore, we analyzed the influence of the level of interruption rates (reported by EC2) and waiting time (see Fig 3) in the three regions. We can observe that they correlate, since a user has to wait longer for those SIs whose requests were fulfilled after more than 60 seconds for SI types with medium interruption rate ($\mathcal{SP}$).

One would expect that there should be a high correlation between the running time before interruption (Fig. 4) and the SI level of interruption rate per region, since higher interruption rates should terminate SIs faster and thus reduce the average running time. However, there is a correlation only between the running times for $\mathcal{F}$ with low and $\mathcal{SP}$ with medium levels of interruption rate. Region $\mathcal{V}$ reported lower running times than $\mathcal{F}$, but the running time behavior of $\mathcal{V}$ was closer to the one of $\mathcal{SP}$.

VI. Related work

A broad range of research explores the use of SIs for reliable computing and introduces a cost/reliability trade-off to overcome the (un)reliability of SIs. We discuss related work separately for approaches based on simulated and real EC2 Cloud environments.

A. Bidding strategies to mitigate SI interruption rates

The long term spot pricing history was exploited by several researchers to estimate the spot price in the near future based on existing techniques. For example, Tang et al. [20] used Markov decision processes, Khandelwal et al. [5] worked with Random forest, and Chhetri [6] used time series forecasting. Other researchers developed their own prediction models [3], [4], [11], [10]. All of these approaches proposed a bidding strategy that *i*) mitigates the SI interruption rate due to *low-bid-price* and that *ii*) reduces the overall cost. Several researchers developed bidding strategies and applied them to achieve cheaper, robust and more reliable large-scale scientific application execution [8], [12], [14].

Although bidding strategies based on price can mitigate the risk of SI interruptions, most of these works evaluate their techniques based on a simulated and simplified spot market environment by considering the bid price only. In contrast, our work is based on the real EC2 spot market and uncovers that most interruptions occur due to capacity reclaiming and not due to the bid price behavior. Bidding strategies should thus consider other parameters as well, such as waiting time for obtaining SIs and random interruptions due to capacity problems.

B. Performance Analysis of SIs in real EC2 environment

Only few papers have conducted empirical studies of the real Amazon's spot market, limited to a single feature of the SI life-cycle, or comprising multiple features but without experimental evaluation.

Mao and Humphrey [21] evaluated the waiting and deployment time together as part of the SI startup time in EC2. They reported that the startup time of SIs is longer than for equivalent On-Demand instances without separating the waiting and deployment time. The authors have considered only a few SI types without taking interruption rates into account.

Similar to our spot life-cycle model, Wang et al. [22] introduced a predictor that advices users on how many SIs to request. The authors claimed that the interruption rate due to low capacity can be neglected which is very much against our findings as shown in Table VII and in EC2 reports [23].

VII. Conclusion and future work

EC2 SIs offer an interesting alternative to reliable fixed price Cloud resources at low costs but at the risk of unreliable instances that at times may not be available. The main motivation is to explore the behavior and performance of SIs for effective usage in scientific computing. We extensively tested SIs by requesting a large number of different SI types covering 3 different regions in North America, South America and Europe. We focused on those SIs that are predominantly used for scientific computing which comprises general purpose, compute optimized and storage optimized instances. We described the different phases and important events of the life cycle for SIs. Our broad study of 3840 EC2 spot requests based on a model for the life cycle of a SI resulted in a variety of findings. The reliability (interruption) of a SI type also influences running time before interruption of this type. If the request for an SI of a specific type is not fulfilled within 4 seconds then is very unlikely that this request will be fulfilled within one minute. This should be important information for Cloud services such as provisioning or scheduling systems that require some Cloud resources with a time limit. SIs are notably reliable in the first 20-30 minutes after deployment even for SI types with medium level of interruption rate. Biding strategies are likely to fail for SIs with medium level of interruption rate which are mostly interrupted due to lack of capacity and not due to low bid price. We also found out that 78% (40%) of all SI requests with low (medium) level of interruption rate behaved largely identical as On-Demand resources for the first four hours of their runtime.

We are currently developing a new resource manager and scheduler for scientific workflows that target both fixed price and volatile resources. We will exploit the results of this study in order to build performance models for SIs that guide both of these tools to effectively use the spot market.

References

[1] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "A performance analysis of EC2 cloud computing services for scientific computing," in *Cloud Computing*. Springer Berlin Heidelberg, 2010, pp. 115–131.

[2] Amazon EC2. Spot Instance. Accessed on 5^{th} March. [Online]. Available: https://aws.amazon.com/ec2/spot/

[3] B. Javadi, R. K. Thulasiramy, and R. Buyya, "Statistical modeling of spot instance prices in public cloud environments," in *IEEE International Conference on Utility and Cloud Computing*, 2011, pp. 219–228.

[4] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir, "Deconstructing Amazon EC2 spot instance pricing," *ACM Trans. Econ. Comput.*, vol. 1, no. 3, pp. 16:1–16:20, Sep. 2013.

[5] V. Khandelwal, A. Chaturvedi, and C. P. Gupta, "Amazon EC2 spot price prediction using regression random forests," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1–1, 2017.

[6] M. B. Chhetri, M. Lumpe, Q. B. Vo, and R. Kowalczyk, "On estimating bids for Amazon EC2 spot instances using time series forecasting," in *IEEE International Conference on Services Computing*, 2017, pp. 44–51.

[7] B. Kamiński and P. Szufel, "On optimization of simulation execution on Amazon EC2 spot market," *Simulation Modelling Practice and Theory*, vol. 58, pp. 172 – 187, 2015, special issue on Cloud Simulation.

[8] W. Voorsluys and R. Buyya, "Reliable provisioning of spot instances for compute-intensive applications," in *IEEE Int. Conf. on Advanced Information Networking and Applications*, ser. AINA '12, 2012, pp. 542–549.

[9] Y. Song, M. Zafer, and K.-W. Lee, "Optimal bidding in spot instance market," in *Proceedings IEEE INFOCOM*, March 2012, pp. 190–198.

[10] A. Andrzejak, D. Kondo, and S. Yi, "Decision model for cloud computing under SLA constraints," in *IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, Aug 2010, pp. 257–266.

[11] R. Wolski, J. Brevik, R. Chard, and K. Chard, "Probabilistic guarantees of execution duration for Amazon spot instances," in *International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '17, 2017, pp. 18:1–18:11.

[12] R. Chard, K. Chard, R. Wolski, R. Madduri, B. Ng, K. Bubendorfer, and I. Foster, "Cost-aware cloud profiling, prediction, and provisioning as a service," *IEEE Cloud Computing*, vol. 4, no. 4, pp. 48–59, July 2017.

[13] M. Lumpe, M. B. Chhetri, Q. B. Vo, and R. Kowalczyk, "On estimating minimum bids for Amazon EC2 spot instances," in *International Symposium on Cluster, Cloud and Grid Computing*, 2017, pp. 391–400.

[14] D. Poola, K. Ramamohanarao, and R. Buyya, "Enhancing reliability of workflow execution using task replication and spot instances," *ACM Trans Auton Adapt Syst*, vol. 10, pp. 301 – 321, 2016.

[15] T. P. Pham, J. J. Durillo, and T. Fahringer, "Predicting workflow task execution time in the cloud using a two-stage machine learning approach," *IEEE Trans. on Cloud Comp.*, vol. PP, no. 99, pp. 1–1, 2017.

[16] S. Ostermann and R. Prodan, "Impact of variable priced cloud resources on scientific workflow scheduling," in *Euro-Par 2012 Parallel Processing*. Springer Berlin Heidelberg, 2012, pp. 350–362.

[17] M. A. Rodriguez and R. Buyya, "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds," *IEEE Transactions on Cloud Comp.*, vol. 2, no. 2, pp. 222–235, April 2014.

[18] ——, "Budget-driven scheduling of scientific workflows in IaaS clouds with fine-grained billing periods," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 12, no. 2, pp. 1–22, May 2017.

[19] R. Mathá, S. Ristov, and R. Prodan, "Simulation of a workflow execution as a real cloud by adding noise," *Simulation Modelling Practice and Theory*, vol. 79, pp. 37 – 53, 2017.

[20] S. Tang, J. Yuan, and X.-Y. Li, "Towards optimal bidding strategy for Amazon EC2 cloud spot instance," in *IEEE International Conference on Cloud Computing*, ser. CLOUD '12, 2012, pp. 91–98.

[21] M. Mao and M. Humphrey, "A performance study on the vm startup time in the cloud," in *IEEE Fifth International Conference on Cloud Computing*, 2012, pp. 423–430.

[22] C. Wang, Q. Liang, and B. Urgaonkar, "An empirical analysis of Amazon EC2 spot instance features affecting cost-effective resource procurement," in *ACM/SPEC on International Conference on Performance Engineering*, ser. ICPE '17, 2017, pp. 63–74.

[23] Amazon EC2. Spot Instance FAQ. Accessed on 5^{th} March. [Online]. Available: https://aws.amazon.com/ec2/faqs/