International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 9, No.5 (Sep-2020)

http://dx.doi.org/10.12785/ijcds/090504

An Approach to Reduce Cloud Spot Instances Cost

Ali Jassim Hasan[1] and Mustafa Hammad[1]

[1]*College of IT, University of Bahrain*

Received 20 Oct. 2019, Revised 20 Jun. 2020, Accepted 31 Jul. 2020, Published 1 Sep. 2020

Abstract: The cost reduction is one of the attractive features offered by the cloud. Spot leasing is one way to reduce the cost. Spot leasing is done by leasing the unused excess instances with low price. On the other hand, spot instances are facing risks that minimize their reliability and desirability. Risks including instances reclaiming and dynamic price changing. Minimizing the risks associated with the spot leasing is going to help to increase the utilization of the spot instances, which in turn is going to attract more users. In this paper, a framework has been proposed to mitigate the instances reclaiming risk while reducing the leasing cost as possible. This is done by monitoring many markets and hopping between instances. The proposed framework has been evaluated through simulating using randomly generated data and actual data collected from Amazon web services. The proposed framework recorded 9% to 42% of cost reduction compared with the actual cost.

Keywords: Cloud, Spot Instances, Cost Reduction, Migration, Hopping.

1. Introduction

The cost reduction is one of the main reasons for attracting clients toward the cloud. The services provided by the cloud are considered cheaper if compared with the costs of traditional services. Cloud service providers, such as Amazon, Microsoft and Google are maintaining large data centers around the world. Those data centers are providing internet services, computing resources and cloud storage options with relatively low cost. Moreover, cloud providers are offering payment models, such as the pay-as-you-use model for on-demand resource allocation. From a customer point of view, cost reduction came from leasing the infrastructure resources with low cost, rather than paying an upfront payment to purchase and build such infrastructure.

There are various types of services offered by cloud providers to cover the customers' requirements. The services can be classified into several categories. Firstly, the Software as Service (SaaS). An example of SaaS is Google mail services, which can be leased as an independent service to the customers. In this service type, the customer only receiver the leased service, no access given to the host or the operating system that is used to provide the service. The second service type is Platform as a Service (PaaS). In this type, the customer has the ability to lease an operating system, such as windows or Linux. There will be no need for the customer to bother himself by dealing with the hardware configuration. The third type of services is Infrastructure as a Service (IaaS). In this service type, the customer has the option to choose the hardware infrastructure. Moreover, the customer will choose the required operating system for the leased hardware.

On the other hand, there are different plans specially made for infrastructure leasing. Each leasing plan has different specification compared to others. Firstly, the reserved plan gives the customer the ability to reserve an infrastructure for a long, uninterrupted period with low cost, which requires to sign a contract. This plan is suitable for certain purposes which cannot tolerate being interrupted and need to be reliable all the time. The second leasing plan is called on-demand. It has a higher leasing cost, but with no contract required. This plan is recommended to be used for the jobs that require a short amount of time with no interruption. As the case with an extra server needed to be added to handle an extra load, and then released after the extra load is finished. The last leasing plan is called spot-leasing. The spot instances can be leased at a cheaper price compared to other plans. This type of leasing was designed to be used by the processes that are time-insensitive and can tolerate getting interrupted. The priority of leasing instances goes to customers with reserves plan, second priority goes to on-demand. The spot leasing came from the extra servers that not reserved nor used for on-demand customers. Those extra servers are leased by relevantly cheap price from a bidding system which keeps changing by the effect of supply and demand. If the number of the unused servers are low and there is a high demand on such devices the spot instant cost will become high. This is making the spot instances cost changing dynamically.

Spot instances are not considered as a reliable option. There are several reasons behind that. First, the dynamic changing for the price. the Second is the recoverability option, which allows the cloud platform to stop the currently running processes and then release the instance from the customer and lease it to another. Moreover, the leasing priority that leaves the spot instances at the end of the list.

Making the spot market reliable is considered a goal that needs to be achieved. if that happened, the total cost of

E-mail: ajasim@uob.edu.bh, mhammad@uob.edu.bh

cloud cost will be reduced even more. The spot instances than would be able to be used for more application.

Currently, there is no mechanism that is designed to provide the spot instances with the required reliability against recoverability, and at the same time, to maintain the leasing cost as low as possible to resist the price changing.

This paper proposes a framework that can be integrated with the virtual machine operating system that is hosted within the spot instances. The framework main task is to continuously monitor the prices of spot instances from several regions. The framework lists the cheapest suitable instances that can handle the running task based on its hardware resource. The generated list is going to be used to make quick decisions against the price changing and reclaiming risks. The proposed framework is going to provide the required reliability to the spot instances. This is done by hopping from one instance to another when any platform reclaim notification is detected. Moreover, since the cloud platform is charging per hour, the framework is going to check the instances price on an hourly basis. If a cheaper instance has found, then the running virtual machine is going to be migrated to the cheaper instance, and this is used to maintain the leasing cost as minimum as possible. The migration process might be within the same cloud of inter-cloud depending on the price and availability of the instances.

The remainder of this paper is structured as follows sections. Section 2 presents a summarization for the literature review. Section 3 discusses a background to the research topic. In Section 4, the proposed framework has been introduced. Then, the Framework evaluation is shown in Section 5. The limitations in Section 6. Finally, Section 7 reflects the conclusion and future work.

2. LITERATURE REVIEW

Cloud computing is considered a promising technology trend. Many companies today are seeking to utilize the benefits offered by the cloud. Benefits, such as cost reduction, reliability, scalability and more. On the other hand, there are other companies trying to avoid dealing with the cloud because of the associated disadvantages. Internet demanding, data privacy, limitation of configurations and security are examples that might affect companies' decisions whether to join the cloud or to avoid dealing with it. Works in [1] [2] [3] presented the benefits that can be obtained from the integration with the cloud, along with the disadvantages and challenges that came associated with such integration process. Hosseini et al. [3] showed how did cloud migration help in reducing the total implementation costs. In contrast, Ryan et al. [4] highlighted the main barriers for cloud embracing, that are the lack of trust between the clients and service providers, accountability, auditability and security.

The cloud basically is a data center, accessible through the internet and containing a pool of resources. Resources that can be leased when needed, which reduced the total cost required to launch a service and maintain it up and running. The cloud service providers are offering many services. The services can be classified as Infrastructure as Service (IaaS), Application as a Service (AaaS) and Software as a service (SaaS). Zhang at el. [5] presented some technologies that are used within the cloud provider internal infrastructure, along with some services that are Offred to the clients. Moreover, a comparison between the cloud providers has been presented, shows some of the commercial products.

Utilizing the spot market infrastructure is considered an important challenge that needs to be solved. Zhang et al. [6] presented a dynamic allocation mechanism for spot market resources based on choosing the best suitable hardware for the required tasks, that can help to reduce the total cost of leasing. Work in [7] presented a cost-aware provisioning system that works on reserving the best suitable server available that matches the needs. Menache et al. [8] presented an algorithm that can be used to dynamically allocation proper resources for batch jobs. The algorithm is taking the price of the on-demand and spot market resources as a factor before deciding to go with which option. Xin et al [9]. Proposed a cloud scheduler that can reduce the total cost of resources leasing. Moreover, they successfully maintained a web server up and running with no interruption using the spot market. Work in [10] presented a mechanism to monitor the history of spot market prices and help in choosing the cheapest.

There are several attempts to migrate between different cloud resources. Works in [11], [12] studied the effects of VM migration between different instances. On the other hand, Works in [13] studied the instance migration based on network traffic minimization. In [14], a proposed solution to monitor the network traffic exchanged between the leased services has been introduced. The proposed mechanize tries to minimize the network latency between the services. The proposed system is working on migrating those services that need to communicate with each other into a close location. The proposed system has been simulated. The simulation results showed that inter could traffic has been reduced by 25% to 60%. Shastri and Irwin in [15] presented a prototype resource container that keeps monitoring the market price. that container can hop between instances based on the price changing. Moreover, the proposed prototype has been implemented. The implementation shows that the prototype has the ability to reduce the cost of provisioning. A novel cloud federation system has been proposed in [16]. That system in monitoring the market price changing. When a new service is needed, the proposed system is going to select the cheapest provider. Moreover, with any price-changing detected, the system rearranges the location of the services based on the lowest price available. The proposed system has been simulated. The simulation results showed that the system can lower the provisioning price using multiple cloud providers.

Comparing our proposed work in [17] with the works presented in [6] [8] focuses on selecting the appropriate resource based on choosing the best suitable hardware. The work proposed in this paper takes the leasing cost into consideration in addition to the proper hardware resources. Comparing our proposed framework to the work shown in [7], our framework keeps monitoring the cost even after finishing the selection and leasing process. That kept the framework aware of any cost changes. Moreover, the proposed framework keeps checking for lower leasing costs in hourly bases, if any cheaper price has found, then the proposed framework hops to the cheaper in order of reducing the leasing cost. The work proposed in [9] focuses on maintaining a server up and running on a spot instance. The migration trigger in this scheduler was based on the cost, while our framework was taking the cost and the resource utilization percentage into consideration. The prototype presented in [15] affected by any cost chancing, even if that change was for a short period. That is going to increase the number of hops taken by the virtual machine. If compared to our proposed framework, the migration is happening on an hourly basis. This is going to filter any spikes that might affect the leasing costs. If the cost changing has lasted for more than one hour, then the proposed framework is going to consider it and going to do the hopping if needed. Furthermore, the costs shown in this paper are calculated based on an actual leasing cost, while the other references are not showing such readings.

3. BACKGROUND

In this Section, a background about cloud platform is presented, including the Regions, Availability zones, and instance types and models. The following Subsections are structured as follows. First, in Subsection 3.1, a general idea about the instance types and models is presented. Then, Subsection 3.2 talked about the regions and the availability zones within. The cloud platform pricing mechanism is presented in Subsection 3.3. Finally, in Subsection 3.4, the risks associated with the spot leasing are highlighted.

A. Instance types and models

Cloud providers are classifying their instances by type and models. Instances types, such as general-purpose, compute-optimized, memory-optimized, storage-optimized and more. Each type is optimized to deliver a certain level of service based on its function. Within each instance type category, there are several models based on the resource configuration. As an example, the compute-optimized type is offering instances that are capable of handling intensive computing loads. Within that type category, there are several models, such as C.Large, C.xLarge. Each model has deferent hardware resources which affect the Leasing price [19].

B. Regions

Cloud providers maintaining their hardware in regions distributed around the world. Those regions are geographically separated from each other. When a customer is planning to use the cloud, the customer will be given the ability to choose a suitable region. Mostly, the customers are choosing the nearest region to them or to their customers in order to get better network response. Each region is containing several datacenters called availability zones. The zones are independent, geographically separated from each other's and are connected together using a high-speed network connection. The purpose of using availability zones is to provide redundancy. If one zone gets affected by any kind of service disruption or natural disaster, the other zoned are there to handle the load.

C. Infrastructure pricing mechanism

There are three schemes of leasing for the cloud infrastructure. Those are the reserved, on-demand and the spot leasing. The reserved instances are leased by contract that offering instances with discount percentage. On-demand instances are those leased with no contract but with no discount as well. Both reserved and on-demand are considered non-recoverable, which means that the cloud platform won't reclaim those instances in case of any need. The third leasing scheme is the spot instances. Spot instances are those instances that are available in the data center and not leased by the previously mentioned leasing types. Those instances are considered extra. because of that, the cloud platform allows to lease them with relatively cheap price with the ability to reclaim them when needed. The price cost reduction between on-demand and spot leasing schemes can reach 90% [20].

Fig. 1 shows the percentage of total cost reduction when using spot leasing over on-demand leasing. The figure shows various type of instances. Leasing any instance type using spot leasing scheme has deduced the total leasing cost. The range of total cost reduction using the spot leasing over on-demand for the selected samples was between 70% to 90%, with an average of 79% of cost reduction. As an example, leasing M5.metal instance using spot leasing scheme is going to save about 79% of the leasing cost if compared with the on-demand scheme.

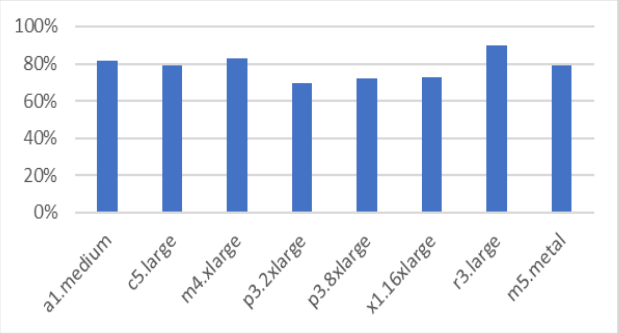

Figure 1. Total cost reduction when using Spot leasing over on-demand for various instance types

The cloud platform is leasing the spot instances using a bidding system. When a customer is requesting to lease a spot instance, the platform will require some details. Those details including the instance type, the region and the maximum price that the customer is willing to bid for that instance. Then the platform is going to check the availability of that instance. If there are plenty of non-leased instances, then the customer will be giving the requested instance. The platform is going to charge the customer with spot price in hourly bases.

D. Spot instances risks

The leasing price is dynamically changing based on supply and demand. If there is a competition of a certain instance type, the spot price for that instance type is going to increase. The customers with the lowest bid price will lose their instances to the one that bid higher. Losing the bidding completion means that the cloud platform is going to reclaim the instance and reassign it to the bidding winner. Losing instances in such that way creates a risk for spot leasing.

On the other hand, the spot price can continue climbing up to become higher than the reserved or on-demand prices. Price changing can work reversibly against the customers, who are trying to reduce the leasing costs by utilizing the spot leasing scheme. Fig. 2 shows the changing of the spot price for M5.24xlarge instance in the market compared with the stability of the on-demand price for the same instance type. As shown in the Figure that the spot leasing cost was stable with a low-cost rate, around 1$ per hour within the period between 28-March 2018 until the 18th of April. Then the price slowly starts to increase until it reaches the on-demand cost, which was around 3$ per hour with the beginning of May 2018. After that, the spot continues climbing up until it reaches around 5.2$ per hour and stays until the end of May. This cost changing is happed due to the high demand for that certain instance model for the shown period.

The cost increment forces the cloud platform to redistribute the instances for the customers with higher priority, which might leave some customers with no instance, or to increase the leasing cost in the best cases. That was an example of the risk that might affect the spot leasing customers.

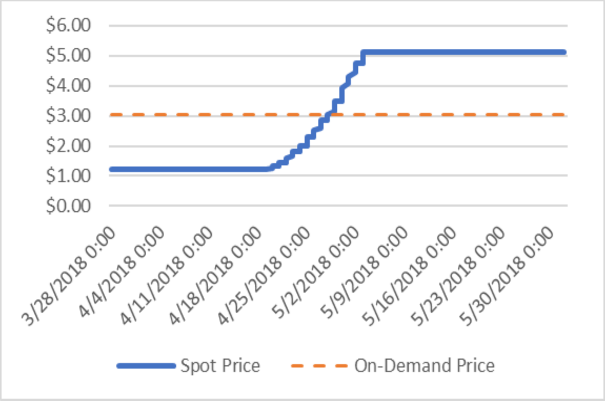

Figure. 2. M5.24xlagre Spot Price compared with on-demand price

4. Proposed Framework

Fig. 3 presents the proposed framework architecture. The framework, in general, works on monitoring the cloud providers and preparing a list of candidate instances. That list is keep updating whenever any change detected. On the other hand, the framework is going to monitor the notification sent by the cloud platform and received by the currently running instance. If the instance received a release/reclaim notification from the current cloud provider, then it uses the prepared list to choose a new instance as a migration target. To find the target instance, the framework starts looking for candidates that can be leased with the minimum cost fees. It starts by looking for a spot instance. If the framework could not succeed to acquire any spot instance, then the framework goes to the second option, which is leasing an on-demand instance. After that, the framework is going to backup the data from the running instance and migrate to the newly selected target. The framework contains three stages including pre-selecting process, the selection process and the post-selection process. The following subsections describe each process in detail.

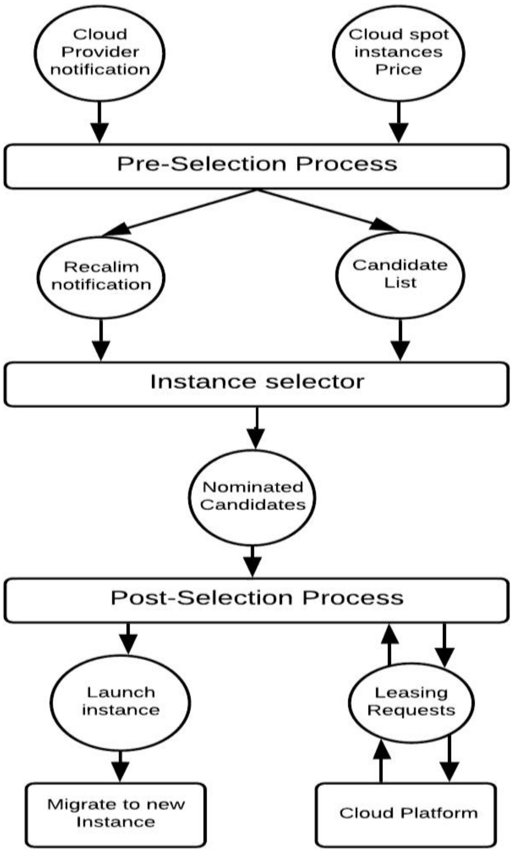

Figure. 3. The Proposed Framework

A. *Pre-selection process*

The spot instances are exposed to be reclaimed by could platform at any moment. If the cloud provider is requesting to take over a certain spot instance, then a notification will be sent to that instance in order to back up the data and stop its process. The instance will be given a time slot to complete its running tasks. After the time slot finished, the cloud provider will interrupt the instance and will take over his resources. The cloud customer is given the option to choose an interruption behaviour that is suitable. There are three behaviours for an interruption which hibernate, stop, terminate.

The pre-selection process responsible of doing two functions. Functions including the notification filtering and cost rate collection. The pre-selection process is continuously monitoring the cloud notifications. On the other hand, it will filter the notifications received from the cloud provider and forwarding the reclaim trigger to the instance selector. On the other hand, the pre-selection process is also monitoring several cloud markets. From different regions, the process is going the check the prices of several instance models. Then it is going to generate a list of candidate instances. Candidates list contain spot and on-demand instances. The options listed must have the equivalent computing power or more compared with the currently running instance. That list is going to be used if migration is required. The pre-selection process will keep updating the candidate list if there is any cost change happened. Generating the list will save the time of looking for an alternative instance, which can be hard to achieve after receiving the reclaim notification.

B. Instance selector

Instance selector is the process that responsible for choosing an instance the going to be used as a migration target. Choosing the best instance is based on several factors, such as the price, leasing type and instance type. The selector is taking the outcome of the price selector (price list) and the cloud listener (reclaim trigger) as an input. The selector is going to wait for any trigger generated by the cloud listener due to instance reclaiming notification. If a trigger received, the selector is going to run a sequence of checking conditions that end with nominating several instances as targets for the migration. Those targets mainly contain spot instances and may contain on-demand instances as a backup. After that, the selector is going to send the nominated spot instances for the post-selection process to place requests to acquire any of them. While the post-selection process is waiting for cloud platform reply, it will keep the preparing for the migration. The preparation includes taking a snapshot for the virtual machine, copying the data to cloud storage and monitoring the time counter. A time counter is used to identify the remaining time from the moment of receiving the reclaim trigger until the actual reclaiming. That time is defined by the cloud provider. If the post-selection process couldn’t successfully lease any spot instance, then the selector is going to choose to lease an on-demand instance based on the service provided by the instance. while using the on-demand the selector will keep looking to any available spot instance that can be leased. If any suitable instance has been found, then the selector is going to migrate from the on-demand to the spot instance.

Fig. 4 shows the interaction between the selector and different instances types and how the selector is going to choose. Before taking any decision, the selector is going to look for the cheapest available instance, wither it was a spot or an on-demand instance. In some cases, such as the increasing of spot leasing cost to become more than the on-demand, the selector might issue a hop request to switch from spot to on-demand and vice versa. This is done dynamically according to cost changing and the availability of the instances.

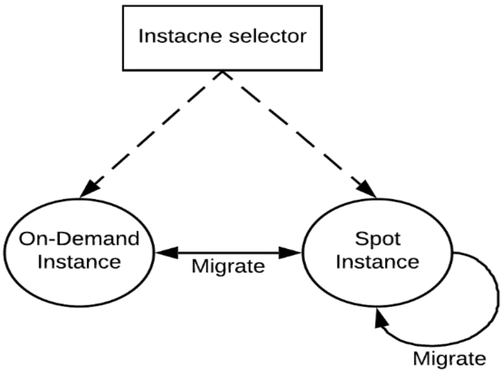

Figure. 4. The decision made by the selector

1) Migration triggers

The instance selector keeps monitoring for many triggers. Triggers including the platform reclaim notification. This notification is sent if the cloud platform is going to reclaim or revocation the instance. After receiving such notification, the instance is given a limited time slot to act before being forced to shut down. In this case, the instance must be migrated immediately. Another trigger that needs to be considered is the instance of resource utilization. If the running service on the instance is utilizing a high percentage of the currently available resources, then the service must be migrated to another instance that has more resources compared with the current. Doing this will maintain a certain level of service performance. One more trigger that is considered important is the current instance price (P-Spot, P-Demand, and P-Bid). Since the P-Spot is dynamically changing due to the supply and demand, there is a chance that it would become high, more than the P-Demand and P-Bid. With such dynamic changing, the scheduler needs to find an alternative instance that is offering the needed recourses at a cheaper price. Those factors are going to be used to determine if there is a need to migrate from the currently running instance or not.

Table 1 shows the triggers that can start the migration process with the action taken by the proposed framework. As an example, when a reclaim notification generated by the cloud platform got received by the proposed framework, then the framework is going to do an immediate migration to the cheapest available instance from the updated list. Such immediate migration is done because of the short time period that is given by the cloud platform. If that period is done and the instance is still running, the cloud platform is going to reclaim the instance by force. This is done by forcing the instance to go to one of the following states, ether terminates or stops or hibernate. On the other hand, if the instance is starving for more resources, then the selector process is going to issue a hop request to migrate the instance to another instance that has more recourses that suits the needs. The last trigger is the leasing cost. Depending on the cost changing, the selector process is going to decide ether to stay or to migrate. If migrate is option is chosen, then to which destination.

2) Instance selecting factors

The migration process is going to require a new instance to be leased before the start of the migration. There are several factors to be considered before selecting the new migration target, such as the price of the currently leased instance compared with other instances from the market. The instance that needs to be selected preferred to have a cheaper price than the current P-Spot. In case that the P-Spot became more that P-Demand, then leasing an on-demand server is going to be considered as a valid option. There is another factor, which is the current instance utilization parentage. This factor is going to assist in determining suitable instances based on the instance type and resources. As an example, assuming that the current instance is having 8 GB of RAM, and the current utilization percentage is 50 %, that will give a glance about the size of the RAM needed by the service. Based on that, the new instance must have at more than 4GB of RAM to be considered as a valid migration target. Table 2 shows how do the selecting factors affect the selector’s decisions. The new instance price and model will be affected based on the current instance price and utilization percentage. The leasing cost is going to play a major role in choosing a suitable destination. If more than one instance is found suitable, then the cheapest is going to be selected. Moreover, if the resource utilization percentage of the currently running instance is going to affect the selection of the new instance. As an example, if the current instance is having 32 GB of RAM, and 70% of that memory is consumed, then the new instance must have more than 32GB.

TABLE 1. MIGRATION TRIGGERS AND ASSOCIATED ACTIONS

Trigger		Action	Details
Cloud Platform Reclaim notification		Immediate Migration to any available instance (spot, on-demand)	Reclaim will be done within a fixed time slot determined by the cloud provider. If there is a ready spot instance then it will be selected, otherwise, an on-demand will be selected
Resource utilization percentage		Find another instance with the required resources then start the migration.	If the current resource percentage has reached to a high limit, then the service must be migrated to another instance with higher resources.
Instance leasing price (P-Spot)	If P-Bid > P-Demand > P-Spot	Check the price every hour and migrate if a cheaper option has found.	The P-Spot keeps changing. if every change is going to be considered as a migration trigger than the service won't be stable.
	If P-Bid > P-Spot > P-Demand	If another spot instance available, then start backup and migration. If not, migrate to an on-demand	In instance will have time to find and lease another spot instance. In case no spot has been found, then leasing an on-demand instance will be a suitable option.
	If P-Spot > P-Bid > P-Demand	Immediate migration	The price of the instance became more than the maximum bidding price.

C. *Post-Selection process*

The post-selection process is including responsible of achieving two functions, which are the instance leasing and the migrating. The post-selection process is going to places requests to launch the selected candidates. The post-selection process is going to be activated by receiving a trigger from the instance selector. The post-selection process as well is going to receive a list of instances which were nominated by the instance selector. The post-selection process is responsible to lease one of the nominated instances. For that, it must provide a request for each instance specifying a bidding price. The cloud provider is checking the customers bidding prices and the highest price customer is going to win. The post-selection process must have cost limitation that is equivalent to ten times of the on-demand cost, which is the maximum bidding price allowed by AWS [21].

After the bidding is done, the post-selection process is going to do the migration. If the selector has chosen an instance, then it will send a trigger to the post-selection process. Then the process is going to launch the selected instance. The post-selection process is also responsible for checking the status of the service of the new instance until the migration procedure is complete. After completing the data transferring and assuring that the service is up and running, the post-selection process is going to release the old instance.

TABLE 2. INSTANCE SELECTING FACTORS WITH THE AFFECTED DECISIONS

Decision	Factor	Details
New Leasing Price	Current leasing Price Vs Market price	New instance price should be the cheaper in the market (spot, on-demand)
	Market pricing	
Instance type and model	Current instance Resources utilization percentage	Based on the utilization percentage, the required resourced for the service will determine the new instance type.

5. PROPOSED FRAMEWORK EVALUATION

The evaluation of the proposed framework has been made in two phases of simulation. In the first phase, randomly generated data has been used to examine the algorithm behaviour. For the second phase, real data collected from amazon AWS has been used. The following subsections describe the process in detail.

A. *Simulation using Random data*

Random data has been created to measure the feasibility of the proposed framework. Table 3 shows the ranges for the randomly generated data that are used in the simulation. The generated data was for 24 hours, for three instances per region, for two regions. The total number of the randomly generated instance was 6 instances. The random range of the spot leasing cost was set to be within 1 to 10$. The on-demand price set to be 5$ and the bidding price set to be 10$.

In order to conduct the simulation, we started by a single availability zone within a region. Then we widened the comparing area by adding the same instance type from different availability zones which belongs to the same region. Moreover, we included the same instance type from multiple regions to the comparison.

TABLE 3. SIMULATION DATA RANGES AND DESCRIPTIONS

Attribute	Description	Range
Time	Simulation period in hours	24
Regions	Number of regions used for simulation	2
Availability Zones	Number of AZ within the region	3
Spot price range	Instance price ranges: Spot price (randomly generated).	1-10$
On-Demand price	On-demand price for the same instance type.	5$
Bidding price	maximum price willing to bid by the customer	10$

TABLE 4. SIMULATION OUTCOME RESULTS

Single Region\ Single Availability Zone

Region	Availability zone	Cost
Region 1	On-Demand	125
	1A	145
	Proposed framework	107

Single Region \ Multiple Availability Zone

Region	Availability zone	Cost
Region 1	On-Demand	125
	1A	145
	1B	151
	1C	129
	Proposed framework	80

Multiple Region \ Multiple Availability Zone

Region	Availability zone	Cost
Region 1	On-Demand	125
	1A	145
	1B	151
	1C	129
Region 2	1A	148
	1B	114
	1C	130
	Proposed framework	52

Table 4 presents the result of the simulation based on the randomly generated data. The table shows that when the hopping range was limited to a single availability zone within a single region. Leasing an instance cost 145$ based on the spot scheme, while it cost 125 on the on-demand scheme for the same instance type. The proposed framework manages to reduce the cost of leasing the instance to become 107$, which is less than the spot by 26% and less than the on-demand by 14%. Furthermore, when the hopping range has been widened to include three availability zones in a single region, the proposed framework has managed to reduce the leasing cost even more than before. The leasing cost reaches 80$ which is also less than the three instances from the three availability zones, with a total cost reduction by 25% compared with the proposed framework cost in the single availability zone. Finally, when the hopping range got widened to include a total of six availability zones from two regions, the proposed framework has scored a reduction in the leasing cost to be 52$, with 35% of leasing cost reduction scored using two regions compared with a single region. This is giving an indicator that widening the hopping area will affect the leasing cost in a reversible proportion.

Fig. 5 shows the hopping ranges and the leasing cost for each range. It also shows how the increasing of the hopping area has given cheaper prices. Fig. 5-A shows the spot price changes within one day for a single instance type with a single availability zone within a region. It shows how did the proposed framework switches between the spot scheme to on-demand leasing scheme and vice versa to find the cheapest price. The figure shows that when the spot leasing became cheaper than the on-demand, the framework will switch to it to reduce the cost. While in case of the increasing of the spot leasing more than the on-demand, the framework is going to choose the on-demand since it became cheaper. Fig. 5-B shows the effect of adding two other availability zones to the hopping range. Adding the extra availability zones has helped in reducing the total leasing cost of the proposed system. Fig. 5-C shows the difference in the cost after adding another region with 3 availability zones, which makes a total of 6 availability zones containing the same instance type located within two different regions. The addition of the extra region has reduced the cost of leasing.

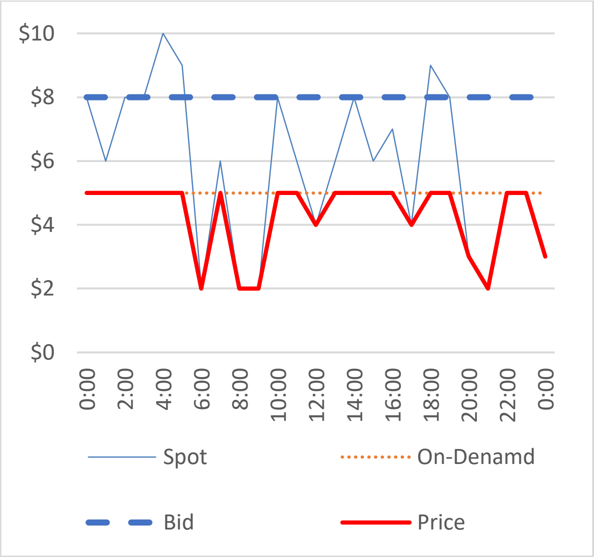

Figure. 5-A Simulation Results for Single Region\Single Availability Zone

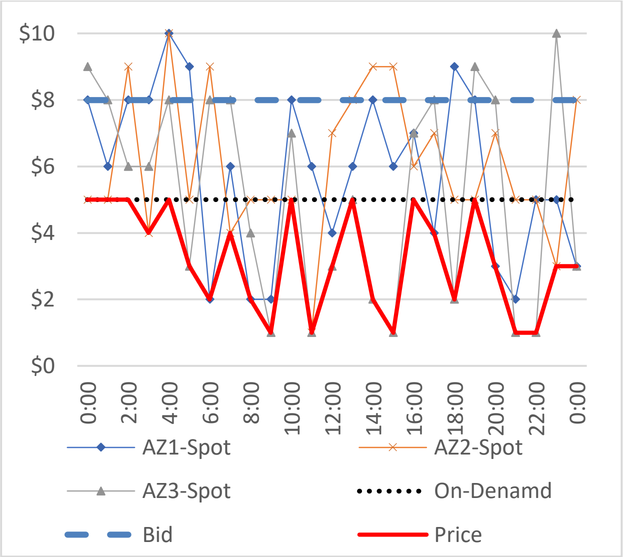

Figure. 5-B. Simulation Results for Single Region\ Multiple Availability Zone

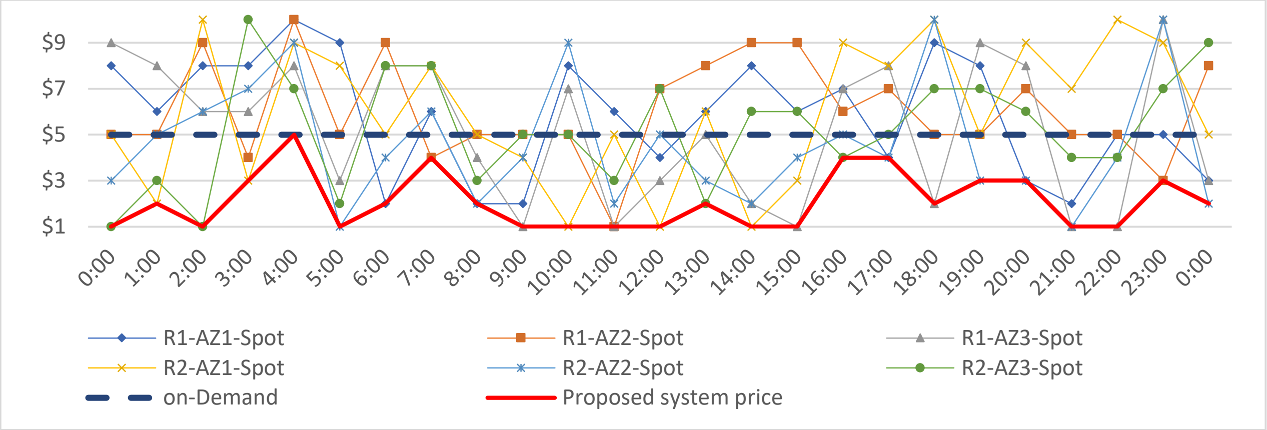

Figure. 5-C. Simulation Results for Multiple Region\ Multiple Availability Zones
Figure. 5. Simulation results using random data

TABLE 5. CASE STUDY PROPERTIES

Instance type	Case study regions	Availability zones	Period		Number of records	Price range ($)		Average price ($)	Price standard deviation ($)	Variance ($)
			Start	End		Min	Max			
M5.24xLarge	CA-Central	1A	3/28/2018	6/1/2018	137	1.21	5.14	3.12	1.67	2.64
		1B	3/28/2018	6/1/2018	140	1.21	5.14	3.35	1.59	2.54
	EU-Central	1A	4/3/2018	6/2/2018	96	1.44	1.62	1.48	0.06	0.004
		1B	4/3/2018	6/2/2018	92	1.44	1.56	1.47	0.06	0.003
		1C	4/3/2018	6/2/2018	92	1.44	1.56	1.47	0.06	0.003

B. Used Dataset

The data used in the simulation process was obtained from a dataset [22]. That dataset was collected and published in 2017 by Supreeth Shastri and David Irwin. The dataset contains the prices of worldwide spot instances as published by Amazon web services (AWS). The dataset is divided into folders. The top-level folders were named to reflect the date that such data was produced by AWS. Within each folder, there are sub-folders that are representing the geographical regions. Within each region folder, the data related to that region instances can be found. Each instance name is reflecting the instance type along with its availability zone. Each file contains data harvested for 60 days directly from AWS with no alteration. The data was obtained using "ec-2-describe-spot-price-history" API command. ec2-spot-prices\2018-may31\eu-central\m5.24xlarge is an example of the dataset hierarchy. It starts by showing the dataset name. Then it shows the date when the data was generated. Next is the region where the instance is located. Finally, the instance type and availability zone for that instance.

The dataset files contain a set of recodes. Each record is generated due to a price change event. The record is composed of seven-tuples that were separated by tap. Tuples attribute including SPOTINSTANCEPRICE: used as an indicator for each row entry. The Price attribute is reflecting the spot leasing cost. The Date/time: the date and time when the entry has been logged. The VM type attribute represents the instance type and model. OS type attribute: This is the type of operating system that is running on the instance. VPC attribute is short term to a virtual private cloud, which is the network that the instance is currently connected. Availability zone attribute: reflect the name of the availability zone where the instance is currently hosted.

C. Case study

Two samples have been taken and analyzed as a case study. The samples are taken from the dataset that contains actual data. Each sample is representing a region that contains multiple availability zones. From those availability zones, a single instance model has been chosen as an input to the proposed approach. Selecting a single instance model has been done to unify the comparison criteria to be the same resources and to measure the actual difference in the prices. Both case study samples are taken from dataset folder named 2018-may31.

The instance that has been selected was from M5.24xLarge model. M5 is the latest model in the M family. The M model is a general-purpose model that is offering a balance between computing, memory and network resources. That makes it a good choice for many applications. M5.24xlarge is coming with 96 virtual CPU, 384 GB of RAM, supports elastic block storage and came integrated with 25 GBPS. More details can be found on AWS website [19].

The first case study sample was CA-Central-1, that is the codename for AWS region located in Canada, that region contains two availability zones 1-A and 1-B. M5.24xLagre samples were taken from each zone. The second sample is EU-Central-1, this is a codename for

Frankfort AWS region. That region contains three availability zones 1-A, 1-B, 1-C. M5.24xLarge samples were taken from those zones as well.

The costs of provisioning were calculated for a single month started from (19-Apr-2018) to (19-May-2018) and the prices are shown in US Dollar. The comparisons were made between the spot prices, on-demand prices (taken from AWS website) and proposed framework prices to show the differences. Table 5 shows the case study samples properties. The table shows the instance type M5.24xLarge, which has been selected as a part of the case study. The table shows the selected regions as well, which were CA-Central and EU-Central. CA-Central includes two availability zones, 1A and 1B. While the EU-Central includes three availability zones 1A, 1B and 1C. The table shows the details about the case study sample. The details include the period when the log file has been started and ended. The number of records in each file is mentioned in the table as well. The minimum and maximum price (cost) range was listed along with the average, standard variation and the variance. As an example, the log file for M5.24xLarge instance from 1A AZ from CA-Central region contains 137 logs. Those logs were recorded in the period between 28th of March to 1st of June 2018. The minimum leasing cost recorded in the log file was 1.21$ per hour and the maximum was 5.14$. The average price was 3.12$ with a standard variation of 1.67$ and with a variance of 2.64$.

D. Simulation using Actual data

Fig. 6-A shows instance M5.24xlarge prices in B1 availability zone within CA-Central region. The figure shows that the spot price has changed to become more than the on-demand instance. The figure also shows the proposed framework price and how did the framework adapt with price changing. After the spot price became more than the on-demand price, the framework chooses to go with the on-demand price since it is lower. This has been done using hopping from the spot instance to the on-demand after increasing of the cost. The figure shows that the proposed price was aligned with the spot at the beginning since the spot was considered cheaper. In 29th of April the spot price has become more than the on-demand, the proposed framework hopped to the on-demand to maintain the price around 3$ per hour. Fig. 6-B shows the difference in the price after adding another availability zone (1A) to the hopping domain. The figure shows that the framework did select the lower price between the two available prices. The figure shows that both instances from AZ 1A and 1B were started with the same price at 19th of April. Then spot cost of the instance located in 1B AZ got increased faster than the same instance in 1A AZ. The proposed system migrated to the 1A since it was lower. Then the 1A cost has increased as well. When 1A hits the on-demand, the proposed system migrated to the on-demand leasing scheme to minimize the price. Moreover, Fig. 6-C shows the price after adding another region (EU-Central) that contains three availability zones (1A, 1B, 1C). Adding an extra region affects the price by noticeable difference. The proposed framework keeps migrating from one spot to another spot instance without the need to lease an on-demand instance.

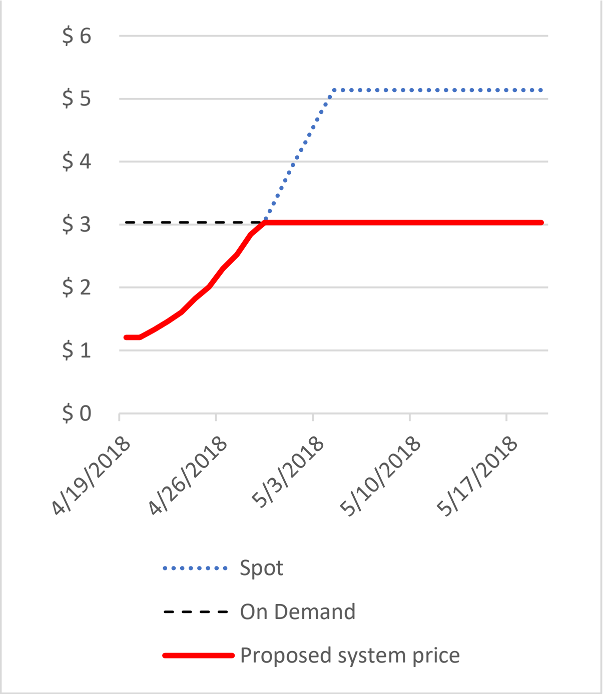

Figure 6-A. Single Region (CA-Central)\Single Availability Zone(B1)

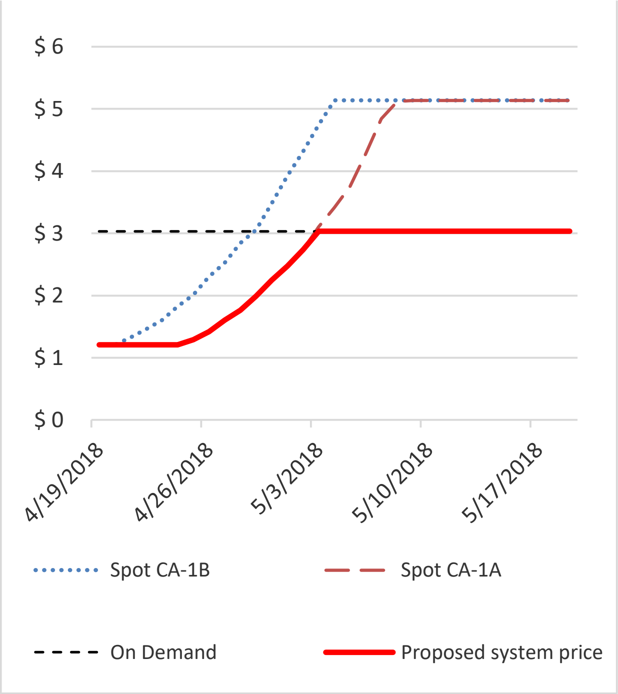

Figure 6-B. Single Region (Ca-Central)\Multiple Availability Zones

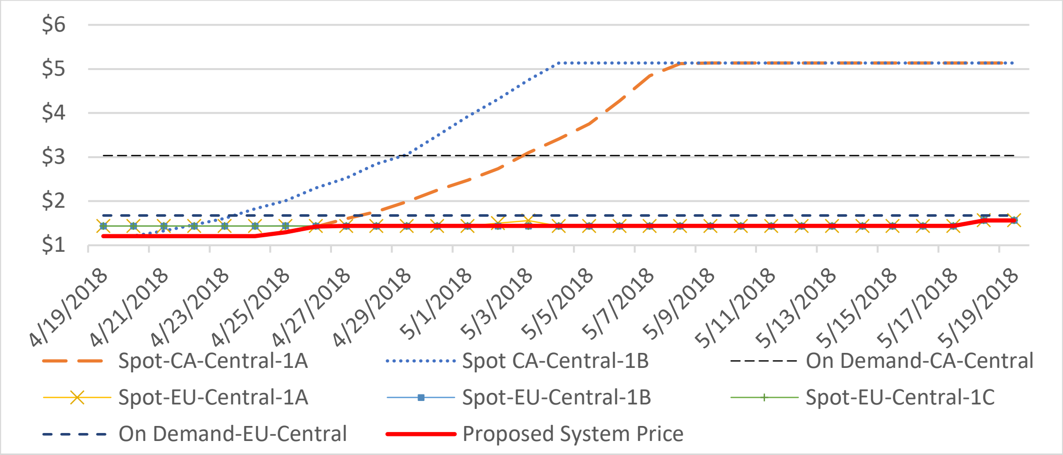

Figure 6-C: Multiple Region\ Multiple Availability Zones
Figure. 6. Simulation Results using Actual data

TABLE 6. TOTAL PROVISIONING COST FOR 1 MONTH

Single region\ single Availability Zone

Region	Availability zone	Cost
CA-Central	On-Demand	2256.85
	1B	2923.17
	Proposed framework	1988.45

Single region \ multiple Availability Zone

Region	Availability zone	Cost
Ca-Central	On-Demand	2256.85
	1A	2532.97
	1B	2923.17
	Proposed framework	1805.3

Multiple Regions \ Multiple Availability Zone

Region	Availability zone	Cost
Ca-Central	On-Demand	2256.85
	1A	2532.97
	1B	2923.17
EU-Central	On-Demand	1243.22
	1A	1080.2
	1B	1075.71
	1C	1075.87
	Proposed framework	1041.66

Table 6 shows the cost of provisioning for M5.24xLarge for one month starting from 19th of Apr 2018 to 19th of May 2018. The table shows how does the hopping range has increased gradually. Starting from a single availability zone from one region. Until reaching multiple availability zones from different regions. The table shows that with the increasing of the hopping domain, the proposed framework price gave cheaper prices. First, when the hopping range was limited to a single availability zone, the recorded spot leasing cost was 2923.17$ compared to 2256.85$ for the on-demand. The difference in the costs indicates the on-demand leasing scheme considered cheaper than the spot leasing cost. This is representing one of the risks that were facing spot leasing. In order to solve this issue, the proposed system has hopped from the spot scheme to the on-demand scheme when the spot price has become higher. Doing this has given the proposed framework the ability to utilize the spot while it is cheaper and hopping to the on-demand when it became cheaper. With that, the proposed framework has leasing cost equal to 1988.45$, which is less than the spot leasing scheme by 32% and less than the on-demand by 12%. On the other hand, when 1A availability zone has been added to the hopping range, the proposed framework recorded a cost of 1085.3$ with a reduction around by 9% compared with the proposed framework cost for single availability zone. Moreover, adding EU-Central region to the hopping range helped in reducing the proposed framework cost to 1041.66$, which is considered cheaper by 42%. This lead to conclude that adding an extra region to the hopping range is going to help to achieve lower leasing costs.

6. LIMITATIONS

There are a few limitations that were encountered. First one was the difficulty in collecting the live spot instances price. Even accessing to instances price history was restricted to only those who are registered with Amazon. Such data is published on the internet, but when it comes to using an API to collect the data, it will require having valid username and password. The second issue was that the other cloud providers are not offering such price history records the same as what Amazon is doing. This is making a comparison between different cloud provider pricing is hard.

Moreover, measuring the impact of resource utilization on the selection algorithm is hard to be conducted in the simulation and need to be implemented on a real instance in order to measure the impact.

7. Conclusion and Future works

The cloud service providers are managing data centers that contain the physical infrastructure and equipment's. According to the supply and demand, there are instances left without being used. The service providers are offering those instances for leasing with low prices, with a chance to be reclaimed at any moment. Utilizing such instances can reduce the total leasing cost even more, but it will be increasing the risk of losing the instances as well. To utilize the extra instances while mitigating the risks, we proposed an approach that is using instance hopping. The proposed instance is monitoring the prices in different regions and migrate the instance if a less price has been found. We run a simulation to test the feasibility of the proposed approach using randomly generated data. Then we run the simulation again using actual data obtained from a dataset. We found that the proposed approach has successfully reduced the cost of cloud instance provisioning. furthermore, the proposed algorithm is going to give better results along with the increasing of the number of regions and the hopping range.

This work can be improved even more if different cloud providers were included. Such a thing can be done by providing a mechanism to crossmatch different instance models based on hardware resources, which in turn is going to help to compare the difference in the prices between the providers. That might open the door to inter-cloud hopping when it came to cut the costs. On the other hand, the proposed algorithm needs to be implemented and tested on a real instance to evaluate its impact on the cost and count the average migrations. Moreover, if instance utilization percentage is known, it will allow hopping between different instance types, and that is going to reduce the cost even farther.

References:

[1] A. Aljabre, "Cloud computing for increased business value," International Journal of Business and social science, no. 1, 2012.

[2] Sajid, M., & Raza, Z., "Cloud computing: Issues & challenges," in In International Conference on Cloud, 2013, December.

[3] Khajeh-Hosseini, A., Greenwood, D., & Sommerville, I, "Cloud migration: A case study of migrating an enterprise it system to iaas," in 3rd International Conference on cloud computing, 2010,.

[4] Ko, R. K., Jagadpramana, P., Mowbray, M., Pearson, S., Kirchberg, M., Liang, Q., & Lee, B. S, "TrustCloud: A framework for accountability and trust in cloud computing.," in In 2011 IEEE World Congress on Services, 2011, July.

[5] Zhang, Q., Cheng, L., & Boutaba, R, "Cloud computing: state-of-the-art and research challenges," Journal of internet services and applications, pp. 7-18, 2010.

[6] Zhang, Q., Zhu, Q., & Boutaba, R, "Dynamic resource allocation for spot markets in cloud computing environments," Fourth IEEE International Conference on Utility and Cloud Computing, 2011.

[7] Sharma, U., Shenoy, P., Sahu, S., & Shaikh, A, "A cost-aware elasticity provisioning system for the cloud," in 31st International Conference on Distributed Computing Systems, 2011, June.

[8] Menache, I., Shamir, O., & Jain, N, "On-demand, spot, or both: Dynamic resource allocation for executing batch jobs in the cloud," in 11th International Conference on Autonomic Computing, 2014.

[9] He, X., Shenoy, P., Sitaraman, R., & Irwin, D, "Cutting the cost of hosting online services using cloud spot markets," in 24th International Symposium on High-Performance Parallel and Distributed Computing, 2015, June.

[10] Yi, S., Kondo, D., & Andrzejak, A, "Reducing costs of spot instances via checkpointing in the amazon elastic compute cloud," in IEEE 3rd International Conference on Cloud Computing, 2010.

[11] J. Zheng, T. E. Ng, K. Sripanidkulchai, and Z. Liu, "Pacer: Taking the guesswork out of live migrations in hybrid cloud computing," Rice University Technical Report, Jan 2013.

[12] R. Bradford, E. Kotsovinos, A. Feldmann, and H. Schioberg, "Live widearea migration of virtual machines including local persistent state," in VEE, 2007.

[13] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in INFOCOM, USA, 2010.

[14] Lu, T., Stuart, M., Tang, K., & He, X, "Clique migration: Affinity grouping of virtual machines for inter-cloud live migration," in 9th IEEE International Conference on Networking, Architecture, and Storage, 2014, August.

[15] Shastri, S., & Irwin, D, "HotSpot: automated server hopping in cloud spot markets," in In Proceedings of the 2017 Symposium on Cloud Computing, 2017, September.

[16] Chi, Y., Cai, W., Hong, Z., Chan, H. C., & Leung, V. C., "A privacy and price-aware inter-cloud system," in 7th International Conference on Cloud Computing Technology and Science (CloudCom), 2015, November.

[17] Hasan, A. J., & Hammad, M, Reducing Cloud provisioning Cost Using Spot Instances hopping, in proceeding of 3ICT2019, ieee (2019).

[18] Hasan, A. J., & Hammad, M, Spot Hopping: Increasing Reliability and Reducing Cost, IJCDS,2020, in press.

[19] "Amazon Ec2 Instance Types," Amazon Web Services, [Online]. Available: https://aws.amazon.com/ec2/instance-types/.

[20] "Amazon Ec2 Spot – Save Up-to 90% on On-demand Prices," [Online]. Available: https://aws.amazon.com/ec2/spot/.

[21] A. AWS, "New Amazon EC2 Spot pricing model," AWS, 13 MAR 2018. [Online]. Available: https://aws.amazon.com/blogs/compute/new-amazon-ec2-spot-pricing/.

[22] S. Shastri, "Amazon EC2 spot price traces," 28 Aug 2018. [Online]. Available: Amazon EC2 spot price traces.

Ali Jasim is a Demonstrator in the Department of Computer Engineering at the University of Bahrain. He is an Ms.C student in the college of IT at The University of Bahrain. He holds a BS.C degree in Computer Engineering from University of Bahrain, 2009. His research interest include cloud computing, IoT and machine learning.

Mustafa Hammad is an Associate Professor in the Department of Computer Science at the University of Bahrain and Mutah University. He received his Ph.D. in Computer Science from New Mexico State University, USA in 2010. He received his Masters degree in Computer Science from Al-Balqa Applied University, Jordan in 2005 and his B.Sc. in Computer Science from The Hashemite University, Jordan in 2002. His research interests include machine learning, software engineering with focus on software analysis and evolution.