

Service Provisioning of Spot Virtual Machines based on Optimal bidding in Cloud Computing



By: Saman Safdar

288-FBAS/F09/MSSE

Supervised by:

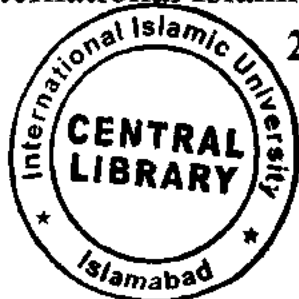
Saeed Ullah

Assistant Professor

Federal Urdu University for Sciences And Technology, Islamabad

Department of Computer Science and Software Engineering
Faculty of Basic and Applied Sciences
International Islamic University Islamabad

2015



Accession No. TH-14748 K/A/

MS
006.78
SAS
C2

- Cloud Computing
- Multimedia systems
- Files

A Thesis Submitted To
Department of Computer Science and Software Engineering,
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad
As a Partial Fulfilment of the Requirement for the Award of the
Degree of Master in Software Engineering.

Dedication

I would like to dedicate this research work to

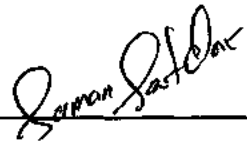
The Holiest man ever born,

PROPHET MUHAMMAD (PEACE BE UPON HIM)

And to my Parents and Teachers.

Declaration

I hereby declare that this Thesis "*Service Provisioning of Spot Virtual Machines based on Optimal bidding in Cloud Computing*" is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.



Saman Safdar

288-FBAS/F09/MSSE

Department of Computer Science and Software Engineering
Faculty of Basic and Applied Sciences
International Islamic University Islamabad

Date: 29-07-2015

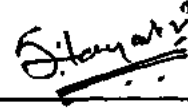
It is a certificate that we have read the thesis submitted by Ms. Saman Safdar and it is our decree that this dissertation of satisfactory standard to certify its acceptance by the International Islamic University Islamabad, for MS degree in Software Engineering.

COMMITTEE

External Examiner

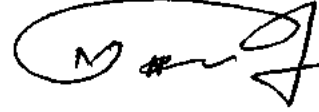
Dr. Malik Sikandar Hayat Khiyal
Professor,

Faculty of Computer Science,
Preston University, Islamabad



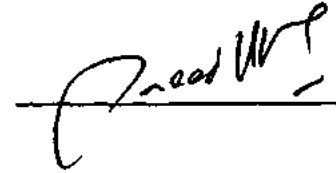
Internal Examiner

Muhammad Nasir
Lecturer (DCS&SE)
IIUI



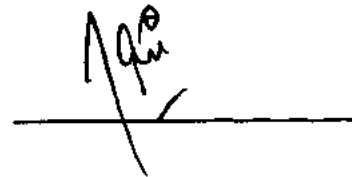
Supervisor

Saeed Ullah
Assistant Professor
Federal Urdu University Islamabad



Co-Supervisor

Zakia Jalil
Lecturer
IIUI Islamabad



Acknowledgement

First of all I am obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this research. I am thankful to many learned people, for their help, guidance and sincere cooperation without which this achievement was not possible. I would like to express my sincere and deepest gratitude to: Mr Saeed Ullah, for his supervision, patience and guidance. His invaluable help of constructive comments and suggestions throughout the Thesis work have contributed to the success of this research. He has been very generous in his support and cooperation. Not forgotten, my appreciation to my co-supervisor, Ma'am Zakia Jalil for her valuable suggestions, and support regarding completion of this thesis.

Besides, special thanks to my friend Sahar Arshad for her constant help, valuable comments, moral support and suggestions were obliging and worthwhile.

Lastly, I would like to thank my family for all their love and encouragement. Particularly, I am extremely grateful to my mother Ulfat Begum and my sister Saadia Safdar for encouraging me. I can't forget the support and endurance of my elder brothers Shahzad, Shahbaz and Sajjad throughout these years who helped me in every way. Without their unconditional support, this work would not have been accomplished.

Thank You!

Table of Contents

Chapter 1 Introduction.....	1
1.1 Overview.....	1
1.2 Problem statement.....	2
1.3 Motivation.....	2
1.4 Research question.....	3
1.5 Research objective.....	3
1.6 Research Methodology.....	3
1.7 Proposed Solution	4
1.8 Dissertation Outline	5
Chapter 2: Background.....	7
2.1 Cloud computing	7
2.2 Deployment models.....	8
2.3 Service Models:	8
2.4 Service Level Agreements:	10
2.5 Virtual Machines.....	11
2.6 Pricing Models in Cloud Computing:	13
2.7 Financial Options.....	15
2.8 Black–Scholes model.....	18
2.9 Stochastic volatility models	19
2.10 Model Implementation:	19
2.10.1 Analytic techniques.....	19
2.10.2 Binomial tree pricing model:	19
2.10.3 Monte Carlo Models	20
2.10.4 Finite difference Models	21
2.11 Model Formulation:	21
Chapter 3: Related Work.....	24

3.1	Overview.....	24
3.2	Pricing of the Cloud.....	24
3.3	Financial Option Theory in Literature.....	26
Chapter 4: Problem Definition.....		31
4.1	Introduction.....	32
4.2	Problem Statement.....	33
Chapter 5 :Implementation		33
5.1	Proposed Framework:	33
5.2	Job Runtime Estimation Module	35
5.3	Scheduling Policy.....	37
5.4	Algorithm for VM provisioning and job scheduling.....	38
Chapter 6: Implementation.....		41
6.1	Simulation Experimental Setup.....	41
6.2	Basic Concepts in CloudSim.....	41
6.3	Option Pricing.....	44
6.4	Results and Discussion.....	46
6.4.1	On Demand Resource Provision Scheme	47
6.4.2	Spot VM Resource Provision Scheme.....	48
6.4.3	Resource Provisioning Scheme with Feedback.....	49
6.4.4	Runtime Estimated Resource Provision Scheme.....	52
6.5	Evaluation& discussion.....	50
6.6	Conclusion & Future Work.....	55
References.....		56
Appendix.....		60

Abstract

The use of cloud computing is expanding rapidly. Cloud Computing is revolutionary and scalable methodology for utilizing IT services. However, along with desirable benefits come risks and concerns that must be considered and addressed correctly. With the emergence of cloud computing, computing resources (applications, data storage, platforms, servers) are provisioned on pay-as-you-go model. In the cloud computing, the virtual machine (VM) is one of the most commonly used resource. Along with On Demand and Reserved VMs which are relatively high fixed price. Amazon EC2 provides Spot Instance VM as a economical option which are offered to customers through Bidding. Service provisioning of VMs , i.e. The research has been conducted to find out the optimal bidding procedures for Spot VM and develop a framework to create a win-win scenario both for consumer and Cloud Service Provider. The concept of *Financial options* has been incorporated for acquiring On-Demand VMs for critical jobs. The proposed framework was applied and validated through a java based development toolkit CloudSim.

Chapter 1

1.1 Overview

Cloud Computing is emerging paradigm in the field of Information Technology. It is an extension of parallel computing, distributed computing and grid computing. It provides safe, fast, easy data storage and net computing service run by Internet. People can have everything they need on the cloud. Cloud Computing is the next natural step in the evolution of on-demand information technology services and products.

Cloud computing has grasped the attention of scientific community and business industry towards the provisioning of computing resources as utility and software as a service over a network. Gartner Forecasts worldwide businesses and individuals spending on Cloud Computing services is expected to be \$250 billion in 2017 [1]. Profitability and revenue maximization are the most important goals for any cloud service provider, which can be employed through different pricing models; however, end-users are typically more interested in high satisfaction guarantee through Quality of Service (QoS), cost-effectiveness, usability and availability of cloud resources (Users maximize utility and CSPs maximize profits). Keeping a balance between these two (trade-off) is the most challenging design decision to be made by cloud service providers.

In cloud, provisioning of computing resources is offered in the form of Virtual Machines (VM), being deployed on physical computing nodes/ servers [2]. Cloud data center needs to be efficient and scalable to connect thousands and even thousands of thousands of such physical machines.

Cloud Computing delivers infrastructure, platform, and software (applications) as services, which are made available as payment based services in a “pay-as-you-go” model to customers. These services are respectively mentioned to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [3]. Cloud computing has been build upon the expansion of distributed computing, grid computing and virtualization [4].

Since cost of each task in cloud resources is different with one another, scheduling of user tasks in cloud is not the same as in traditional scheduling methods.

Cloud computing has fascinated and is still attracting a lot of attention both in industry and in academic world offering a multiple range of flexible, on-demand, and highly scalable computing services. The related flexibility and effectiveness of cost make cloud computing a valuable option for organizations in the public and private sectors. Cloud Computing is revolutionary methodology for utilizing IT services. However, along with desirable benefits, come risks and concerns about security, cost, reliability and resource management that must be considered and addressed correctly [6].

Cloud service providers invest highly in establishing data centres and cloud resources but most of time these resources remain under utilization. For increasing business value these unused resources can be offered as spot instances whose price is generally low than peak hours [11]. So instead of being idle these resources can generate revenue for the cloud service providers. The problem for customer is these spot instances are obtained through bidding, so how the bidding should be done to successfully get the instances for job completion and allocated Virtual Machines aren't taken back as prices go up any time.

The proposed research intends to find out issues related with the finding the optimal bidding procedures for spot Virtual Machines in Cloud and develop a framework to create a win-win scenario both for consumer and Cloud Service Provider. The concept of *financial options* has been included for acquiring On-Demand VMs for critical jobs [25].

1.2 Problem Statement

Pricing is based on the type and size of instances, required resources and sometimes it considers the region as well. The traditional business models are fixed cost model.

In cloud computing, no cost model has been developed yet that builds a distinction between variable and fixed cost and specially for Optimal Bidding of Spot VMs

1.3 Motivation:

The aim of this research is to analyze and construct a model that helps to increase business value for the cloud service providers and cost effectiveness for end users.

As cloud computing is an emerging area, a lot of research is going on its economic models, we choose to work on optimal bidding for Spot instances and resource management.

1.4 Research Questions:

1. How can a Cost-efficient and Resilient resource provisioning framework enhance overall Value Optimization for the cloud service provider?
2. How can a Resource Management Model for cloud, based on deadlines and constraints, help resource provisioning to be more efficient?

1.5 Research Objective:

The research is aimed to achieve a win-win scenario for both service provider and consumer by providing low cost services (for consumers) and optimized overall value of available resources (for Cloud service provider). Profit maximization and cost minimization are the main factors in this research. The both results are to be obtained by without violating the Service Level Agreement constraints [3] [5].

1.6 Research Methodology

To explore the problem outlined above, we conducted this research by employing the method outlined in Figure 1.1 and discussed in this section.

Firstly, Cloud Resource Management model is proposed after detailed literature survey comprising on studying existing resources allocation methods, financial options used in cloud pricing. And determination of the parameters for the Model is done through reviewing Amazon EC2 pricing history for Spot instances. [10]

In the next step, implementation of the model is conducted through simulation. The development of the models and those frameworks is evaluated using cloud simulations. Simulation is performed through java based cloud simulator Cloud modeling tool kit i.e CloudSim 3.0 [8].

The model has allowed us to make rules and implement them through performing simulation based scenarios/schemes in cloud environment. Further it enabled us to make policies which are acceptable for perfect bidding. After implementing with different schemes we came to know about best bidding techniques, which when applied to real cloud will increase Business value of the resources of the CSPs.

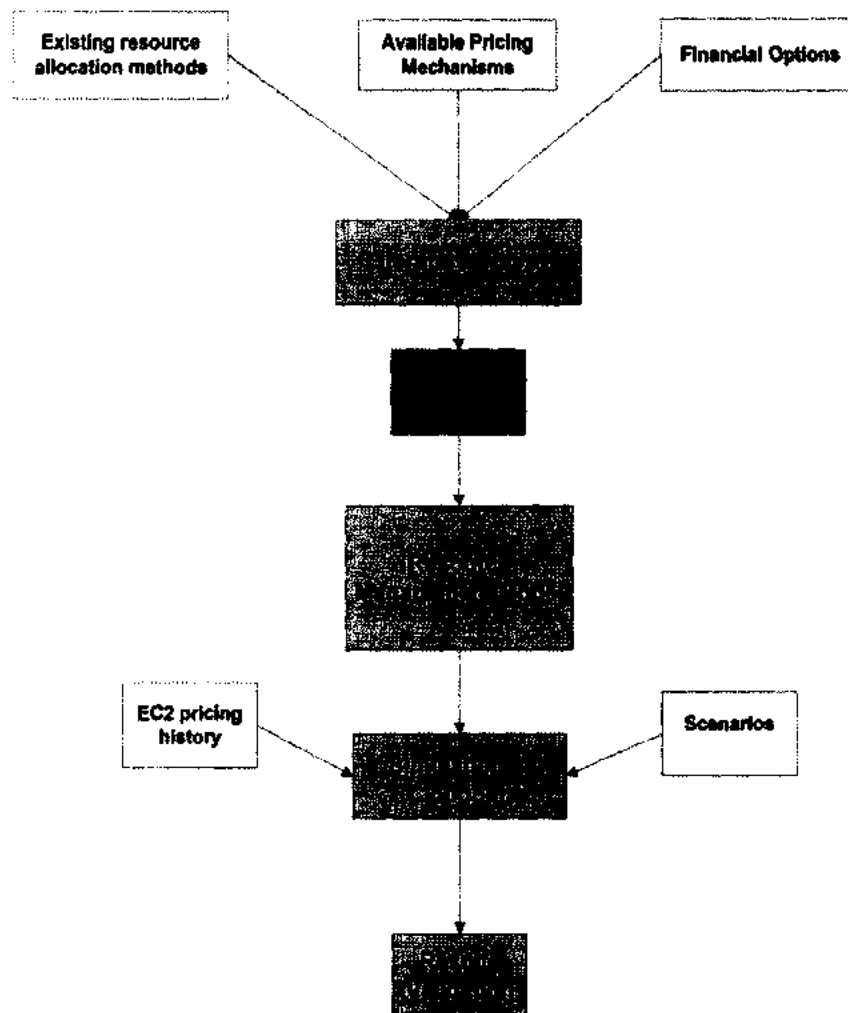


Fig 1.1 Our Research Methodology

1.7 Proposed Solution

We propose a resource management model for optimal utilization of cloud resources and it provides an optimal bidding mechanism. This proposed model helps to increase business value for the cloud service provider and offers cost effectiveness to customer. Basically we included Financial Options for using on demand instances when any job of high priority is encountered and whereas normal jobs are catered through spot instances which results in low

cost. From cloud provider's point of view this model optimizes the overall resource utilization.

1.8 Dissertation Outline

The remaining dissertation is ordered in a manner described in this section. Chapter. 2 establishes and founds the ground for this research by defining basic concepts and terminologies related to cloud computing and virtual machines. Chapter. 3 reviews the existing research related to cloud computing, its economic models, financial options theory in cloud pricing, and resource management models for spot Virtual Machines.

Chapter. 4 narrows down the focus of this research by formalizing the problem statement. Chapter. 5 describes the proposed framework, i.e. the designed algorithm .Chapter. 6 is based on implementation of the proposed model in CloudSim simulator.

Chapter 2

2.1 Cloud computing

Cloud computing has emerged as a new paradigm for delivery of applications, platforms, or computing resources (processing power/bandwidth/storage) to customers in a “pay-as-you-go-model”. According to the National Institute of Standards and Technology (NIST),

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [7].

The Cloud model is cost-effective because customers pay for their actual usage without upfront costs, and scalable because it can be used more or less depending on the customers’ needs. Due to its advantages, Cloud has been increasingly adopted in many areas, such as banking, e-commerce, retail industry, and academy. Considering the best known Cloud service providers, such as Salesforce.com, Microsoft and Amazon, Cloud services can be categorized as:

1. Application (Software as a Service – SaaS)
Examples are NetFlix, DropBox, Gmail, Salesforce.com
2. Platform (Platform as a Service – PaaS)
Frameworks, Databases, as MS Azure, Google App Engine
3. Hardware resource (Infrastructure as a Service – IaaS)
Virtual Machines, Storage

Buyya et al [3] argue that based on the increasingly common perceived vision of computing, it will become the fifth utility (after electricity, gas, water, and telephony), providing the general public with the basic level of computing services used in their daily routines. And it is evident today from common example of smart phone usage of cloud storage, applications etc.

Software systems have been developed from being monolithic one-tier systems to more complex and joint n-tier systems, which accentuate, along with the Internet evolution, the importance and need to have suitable integration technologies to facilitate businesses to communicate over the networks [4].

Therefore, a significant evolution of integration technology has been the development of Service-Oriented Architecture (SOA), which is a pattern used for understanding and maintaining business activities that bridge large distributed systems [3] [11].

2.2 Deployment models:

Generally cloud computing has four deployment models [7]:.

A separate cloud maintained, owned or operated for the use of a single organization is termed as Private cloud. That can exist within the same organization or it can be situated at any other place.

Public Cloud is most commonly used model which is available for general public. Such cloud is placed at the provider's location and can be accessed from anywhere and anytime. Governments, universities and other public organizations use this type of cloud.

Community Cloud is used particular group of people by which share same concern or goal. Community cloud is supervised, controlled, and developed by a group, single organization, third party, or some combination of the three [7].

Hybrid cloud is a combination of any two of the above mentioned deployment models. Users can integrate any two deployment models to introduce some new services and additional benefits.

2.3 Service Models:

2.3.1 Software As A Service (SaaS):

SaaS provides many types of diverse interfaces which are used to access the software applications running at the end of cloud provider's infrastructure. These applications are

managed by cloud providers only and cloud customers cannot get the control over underlying infrastructure of these applications as well as cloud except if certain specific configurations are required by specific users [7].

2.3.2 Platform As A Service:

PaaS provides the deployment of user-created or obtained applications to cloud environment. The resources of consumption like programming languages, services, libraries and tools are provided by cloud providers. Cloud users have limited access to configuration settings of operating system environment but they can manage and configure the deployed applications which facilitate the development of applications. PaaS examples are Microsoft Azure and Google App Engine [7].

2.3.3 Infrastructure As A Service (IaaS):

IaaS provides infrastructure CPUs, storage, networks and other low level resources to their customer in form of virtual machines [32]. The cloud infrastructure is managed by cloud users from an abstract point of view, deploy the software including underlying operation system and monitor the resources whenever they want. Physical infrastructure is still managed by cloud providers only and is not accessible to cloud users.

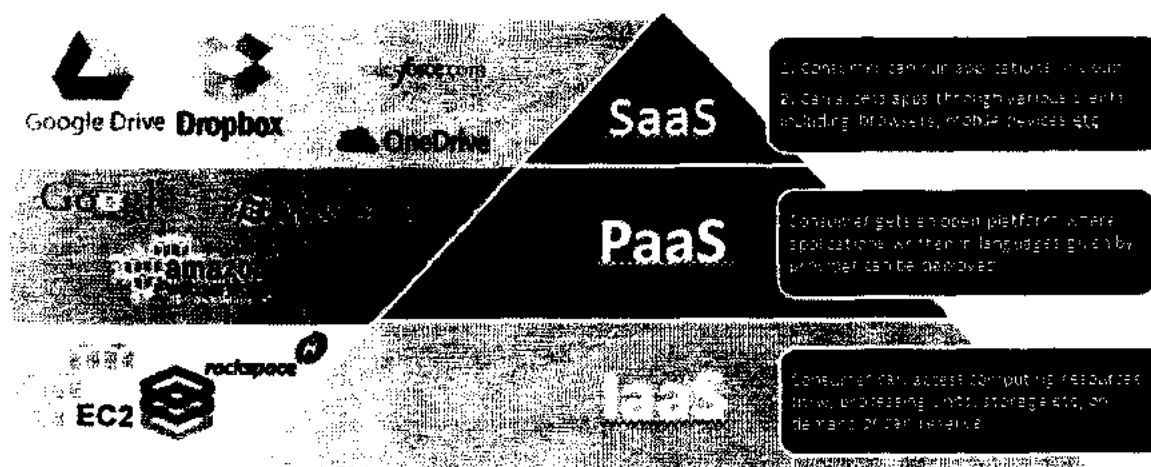


Figure: 1.2 Cloud service Models

Amazon Elastic Compute Cloud (Amazon EC2) offer four types of VM instances: small (S), medium (M), large (L), and extra large (XL).

Cloud Computing is being introduced and marketed with many attractive promises that are appealing to many companies and managers around world, such as reduced capital costs, and relief from managing complex IT infrastructure, to name a few. One of the key challenges that restrain businesses in implementing cloud computing services, even if they found it cost effective that , still is they shift their information and data out of their direct control.

The main concern is how confidentially the Cloud providers keep their information (security) and with which quality they deliver their services (performance). To cope with this challenge, service level agreement (SLA) has been introduced.

2.4 Service Level Agreements (SLAs):

SLA is a binding agreement between the service provider and the service customer, used to specify the level of service to be delivered as well as how measuring, reporting and violation handling should be done.

SLA contract, which is signed by both parties includes Quality of Service (QoS) requirements and penalties in case QoS requirement is not met by providers. However, SLA is not sufficient to ensure Cloud reliability. For example, if a business has critical Web application deployed on Cloud and it fails, thousands of dollars might be lost. Nevertheless, according to most SLA contracts, they only give a penalty as much as a portion of the deployment fee.

Software systems have been developed from being monolithic one-tier systems to more complex and decoupled n-tier systems, which emphasizes, along with the Internet evolution, the importance and need to have suitable integration technologies to enable businesses to communicate over networks [4].

Therefore, a significant evolution of integration technology has been the development of Service-Oriented Architecture (SOA), which is a pattern used for understanding and maintaining business activities that bridge large distributed systems [5]

In Cloud computing, no cost model has been developed yet that builds a prominent distinction between variable and fixed cost. In today's economy, variable cost is the one which is best suited for business strategies, but anyhow distinction is more important to find the accurate cost [6].

Amazon EC2, it has three types of instances which has a number of subdivisions such as small, medium, large and this subdivision includes for all kinds of cloud resources. This instance types are available for several zones. Our research is mainly based on Spot VM but in the following section we will discuss about On-Demand and Reserved as well. These two VM's helps to prove whether Spot VM is a better option for the users by comparing with other VM's [7].

2.5 Virtual Machines

2.5.1 On-Demand Virtual Machine

The On-Demand Instance is reliable for the user who requires running their application for a short period of time and fluctuate workloads without any interruption. On-demand instances are not feasible for long-term jobs and without any limits. [7]

Costs for using On-Demand Instances are much higher than Reserved Instances and Spot Instances because the user should pay the cost per-hour usage fee i.e. the high fixed rate. The user can raise or reduce the serviceable ability of the instances based on their demands. If the user uses the server for a partial hour, that will be calculated roundly as full hour for billing purpose.

2.5.2 Reserved Virtual Machine

Users who are using the reserved instances should pay one time upfront fee payment for each server and it can be purchased for 1 year or 3 years contract only. Once the payment is done for reserving the instance, there is no further obligation. There are three types of instance

which includes Light, Medium and Heavy Utilization Reserved Instances while running these reserved instances, the users can get a prominent discount on the hourly price in Amazon [7].

2.5.3 Spot Virtual Machine:

In On-Demand, the instance price doesn't change (i.e. high fixed cost) and in the reserved instance, the customer should contract for at least 1 year. In December 2009 Amazon EC2(Elastic Compute Cloud) provides the solution for this issue by introducing its "spot instances" pricing system and also it changes a spot price based on supply and demand.[7] [13]

Spot Instances allows to name price for Amazon EC2 computing capacity. Simply bid on spare Amazon EC2 instances and run them whenever the bid exceeds the current Spot Price, which varies in real-time based on supply and demand. The Spot Instance pricing model complements the On-Demand and Reserved Instance pricing models, providing potentially the most cost-effective option for obtaining compute capacity, depending on application.

Spot Instances can significantly lower the computing costs for time-flexible, interruption-tolerant tasks. Spot prices are often significantly less than On-Demand prices for the same EC2 instance types. Additionally, for some distributed, fault-tolerant tasks (like web-crawling or Monte Carlo applications), it will be able to simultaneously accelerate the computational task and reduce its overall cost by opportunistically incorporating Spot Instances [7] [13].

A lot of savings can be generated using spot instances, that can be further invested elsewhere. Because Spot prices are in general much below (recently 86% lower, on average). On Demand prices, you can lower the cost of your interruption-tolerant tasks and, potentially, accelerate those applications when there are many Spot Instances available.

There are four categories of time flexible and interlude tolerant jobs that work better with Spot Instances:

1. **Optional jobs.** The tasks which are not compulsory to run at real time. When Spot prices are low, optional jobs can be ran, and when they rise too high these jobs can be stopped.
2. **Delay able jobs.** These jobs have time deadlines that allow you to be flexible about when you run your computations.
3. **Acceleratable jobs.** These jobs can be speed up by adding additional computing power. You can run Spot Instances to accelerate your computing when the Spot price is low while maintaining a baseline layer of On-Demand or Reserved Instances.
4. **Large scale jobs.** Such large scale jobs might need computing level that one can't access through any other way. If Spot VM is used, the jobs can run cost-effectively thousands or more instances in different AWS regions available around the world.

Amazon is the sole seller in the market of spot instances. It can set a higher price and serve less requests to gain extra revenue. This is so-called market power in economics.

2.6 Pricing Models in Cloud Computing:

Because there is not much information about pricing models on the cloud computing, we are going to discuss them from an economics point of view. Pricing can be seen as a chargeback model which can be based on different businesses. In IT a chargeback model is defined as a user paying for what he/she has used after usage. The model needs to be accurate, auditable, flexible and scalable.

Clouds computing must provide a good pricing model that is beneficial for both parties. It is sometimes hard to find a balance in which both sides agree with the price set. A good pricing model is defined as a price that will carry no loss to neither the provider nor the consumer. From the consumer's point of view a better pricing model is one where they will pay a lower price for the resources requested, while from the provider's point of view, they should not go

beyond the lowest price that provides 0% profit for them as well as increasing the utilization. The consumer's point of view can be summarized as the user satisfaction.

Pricing is based on the type and size of instances, required resources and sometimes it considers the region as well [17]. The traditional business models are mainly based on the fixed cost model due to the large capital investment which measures the product life cycles in years. But in recent days, the product life cycles are measured in terms of months so the consumer largely prefers flexible cost variation models to changing their demands.

Pricing is the method to determine what a company will get in exchange for its product or service. Pricing factors are cost of manufacturing, market place, competition with other organizations, market condition and quality of product. Pricing is an important factor for such organizations which offer products or services; because the price affects many key aspects like customer's loyalty towards that provider, customer's behaviour, and eventually the success of the company. The development of an efficient and effective pricing model can help the organization in achieving the targeted revenues. [19] The price set for each service or product must take into account the maintenance costs, manufacturing costs, competitors in market offering same services/products and the value of the product/service.

The most common pricing model in use for cloud services is pay-as-you-go, which is a static model, and price is set by the service provider. This model doesn't consider customer's concerns except service level agreement (SLA), a negotiation between the provider and the customer regarding the services provided.

Considering the pay as you go model of cloud computing, pricing is one of the critical factors for CPs, offering their services and infrastructure to their clients [14]. There are two main types of cost models for cloud computing: fixed price cost model and variable/ dynamic cost pricing model. Most of the traditional service costs are based on the fixed cost model due to the large capital investment which measures the product life cycles in years. However, in recent days, the product life cycles are measured in terms of months so the consumers largely prefer flexible cost variation models to change their demands.

2.6.1 Static Pricing of Amazon:

For maximum utilization of resources Amazon offer, On-demand, Reserve and Spot Instances .On-demand is hourly based charge; like Google Apps and Amazon EC2.

Amazon simple price calculation formula

$$P = P_{comp} + P_{storage} + P_{in} + P_{out} + P_{tran} \dots\dots\dots(1)$$

P_{comp} is the VM instance price. These include standard, High Memory and High CPU.

$P_{storage}$ is the price charged for storing user data on cloud.

P_{in}, P_{out} is price associated with uploading and downloading the data between different regions of the same cloud.

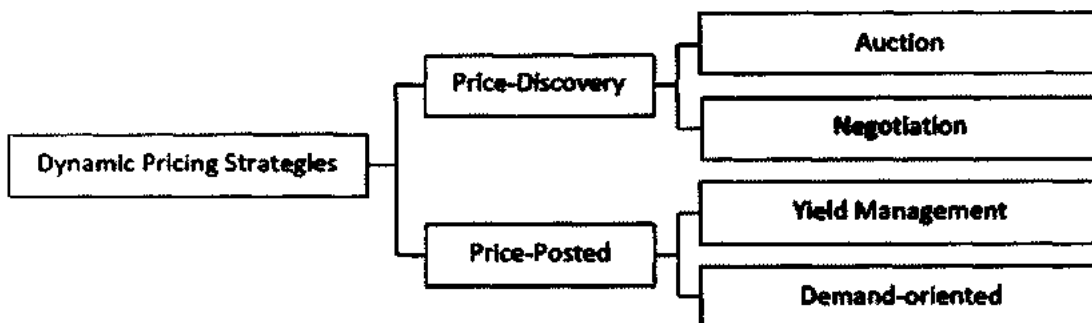
P_{tran} is the price of file operation within a VM.

2.6.2 Dynamic Pricing:

The concept is to utilize unused and spare capacity available in the data centers after fulfilling the demands of the on-demand and reserved instances. These unused capacities are referred as spot instances and are charged based on the fluctuating supply and demand of these spot instances. In cloud computing the provision of leasing technology as a utility is one of the potential opportunities to achieve market-based price by the provider. The concept of auction is the instrument to achieve the potential benefit from the market.

There is no surety of provisioning in On-Demand. Reserve upfront cost is too much. (100-2000 USD)Amazon per VM .Not all cloud providers offers Reserved instances like Rackspace not offers, Google recently offer and only Amazon offers..more over reserve may be rejected by single Cloud Service Provider Issue with spot, only offered by Amazon is preemption issue.

Researchers are investigating different alternatives to the traditional static pricing models. Fig. 2.3 illustrates some of the proposed dynamic pricing strategies:

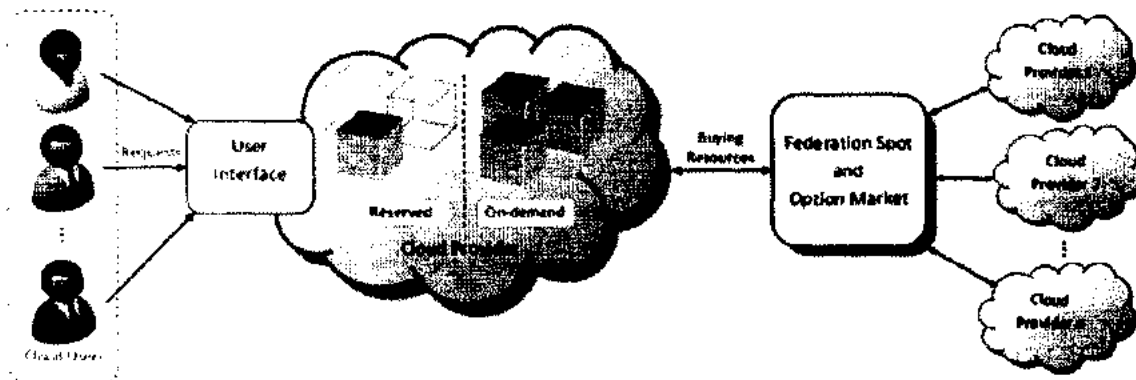


Dynamic pricing strategies in Clouds

Figure: 2.3

This research aims to review and explore another dimension of recently proposed dynamic pricing strategy in the cloud market i.e., pricing cloud computing commodities using financial options. Financial option (future contracts) pricing model gives the right to obtain an instance at a given price, established at the agreement’s stipulation time.

The primary benefit of this model is to make buy options in advance and when required, exercise them to achieve more economies of scale [8]. This just works like an insurance policy where cloud users and providers can hedge against unavailability of resources by paying a premium price in advance to meet the workload requirements in peak hours.



Model elements and architecture

Figure: 2.2

2.7 Financial Options

A Financial Option is an agreement which gives right to the buyer to buy or sell an asset; it is just a right not an obligation. [25]

The basic thought behind an option can be seen in many everyday situations. For example, one would like to purchase a house. But don't have the cash to buy it for another three months, so speaks to the owner and negotiate a agreement that gives an option to buy the house in three months for a price of Rs 20,0000. The owner agrees, but for this option, requires payment of 30,000 as a contract fee. And before the expiry date one can buy the house on agreed price, even if the price of house has increased to 25,0000. But if the current market price has decreased to say 15,0000, one can avoid the purchase on agreed higher price. In this case only the option price will be the loss which was paid in advance.

There are two basic types of options:

"A call option gives the holder of the option the right to buy an asset by a certain date for a certain price."[25]

"A put option gives the holder the right to sell an asset by a certain date for a certain price. The date specified in the contract is known as the expiration date or the maturity date. The price specified in the contract is known as the exercise price or the strike price. American options can be exercised at any time up to the expiration date, whereas a European option can only be exercised on the final expiration date. American options are more practical and widely used, but European options are easier to analyze mathematically."
[25]

The holder has a right to exercise the option, but it does not imply any obligation on him. Options are opposite from forwards and futures, where the holder is obligated to buy or sell the underlying asset." However, forwards and futures are free, where as an investor must pay (cost of option) to procuring an option contract.

The important advantage of buying options in comparison to other future agreements is that it gives the provider the right (not the obligation) to buy resources (outsourced requests) in the

future. Therefore, if the cloud client does not request the reserved instances, the provider will simply let the contract expire without responsibility to buy unnecessary resources. The only cost for providers in such an arrangement is the premium paid at the beginning of the contract. This cost, however, can translate into trust and goodwill by the clients on the provider.

In Financial option theory model for buying VMs, providers transfer the risk of violating SLAs to other providers by buying option contracts and paying option premium. Therefore, sellers of the option contracts must consider the trade-off between the risk and expected profit.

The value of an option can be estimated using different quantitative techniques based on the concept of risk neutral pricing and using stochastic calculus. The most basic model is the Black–Scholes model. More sophisticated models are used to model the volatility. These models are implemented using a variety of numerical techniques. In general, standard option valuation models depend on the following factors:

- The current market price of the underlying security,
- the strike price of the option, particularly in relation to the current market price of the underlying (in the money vs. out of the money),
- the cost of holding a position in the underlying security, including interest and dividends,
- the time to expiration together with any restrictions on when exercise may occur, and
- an estimate of the future volatility of the underlying security's price over the life of the option.

The following are some of the principal valuation techniques used in practice to evaluate option contracts.

2.8 Black–Scholes model:

Fischer Black and Myron Scholes made a foremost achievement by deriving a differential equation that must be satisfied by the price of any derivative dependent on a non-dividend-paying stock. Black and Scholes created a closed form resolution for a European option's hypothetical price. While the ideas behind the Black–Scholes model were ground-breaking and eventually led to Scholes and Merton receiving the Nobel Prize in Economics, the application of the model in actual options trading is awkward because of the assumptions of

continuous trading, constant volatility, and a constant rate of interest. Nevertheless, the Black–Scholes model is still one of the most important methods.

2.9 Stochastic volatility models

“*Stochastic volatility models*” have been created including one developed by S.L. Heston. One major benefit of the Heston model is that you can solve it in closed form, while other “*stochastic volatility models*” do need complex numerical methods.

2.10 Model Implementation:

There are different techniques for model implementation, first the specific model is chosen then any suitable technique is applied.

2.10.1 Analytic techniques

In some cases, one can take the mathematical model and using analytical methods develop closed form solutions such as Black Scholes and the Black model. The resulting solutions are readily computable.

2.10.2 Binomial Tree Pricing Model:

Closely following the derivation of Black and Scholes, John Cox, Stephen Ross and Mark Rubinstein developed the original version of the binomial options pricing model. It models the dynamics of the option's theoretical value for discrete time intervals over the duration of option. The model starts with a binomial tree of discrete future possible underlying stock prices. By constructing a riskless portfolio of an option and stock (as in the Black–Scholes model) a simple formula can be used to find the option price at each node in the tree. This value can approximate the theoretical value produced by Black Scholes, to the desired degree of precision.

However, the binomial model is considered more accurate than Black–Scholes because it is more flexible; e.g., discrete future dividend payments can be modeled correctly at the proper forward time steps, and American options can be modeled as well as European ones..

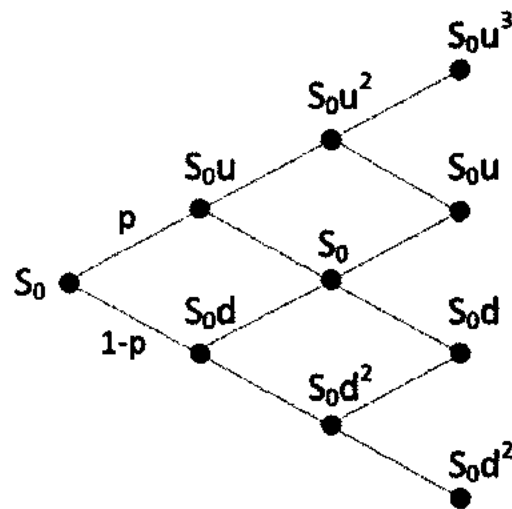


Figure 1.6 Binomial tree for option pricing

Consider the current spot market price is S_0 . S_0 goes to S_{0u} with probability of p and to S_{0d} with probability $1 - p$ at each time step ΔT . Let $T = n \cdot \Delta T$, where T is the option expiration date, then a lattice of spot price movement for $n = 3$ is presented in Figure above. The value of the option can be evaluated for each point at the leaf nodes of tree (time T). The value of the option at starting node can be calculated through a procedure known as backward induction. A call option is worth $\max(S_T - K, 0)$, where S_T is the spot market price for underlying asset at time T .

The Trinomial tree is a similar model, allowing for an up, down or stable path; although considered more accurate, particularly when fewer time-steps are modelled, it is less commonly used as its implementation is more complex

2.10.3 Monte Carlo Models

For many classes of options, traditional valuation techniques are intractable because of the complexity of the instrument. In these cases, a Monte Carlo approach may often be useful. Rather than attempt to solve the differential equations of motion that describe the option's value in relation to the underlying security's price, a Monte Carlo model uses simulation to

generate random price paths of the underlying asset, each of which results in a payoff for the option. The average of these payoffs can be discounted to yield an expectation value for the option. Note though, that despite its flexibility, using simulation for American styled options is somewhat more complex than for lattice based models.

2.10.4 Finite difference Models

The equations used to model the option are often expressed as partial differential equations for example Black-Scholes equation. Once expressed in this form, a finite difference model can be derived, and the valuation obtained. A number of implementations of finite difference methods exist for option valuation, including: explicit finite difference, implicit finite difference and the Crank-Nicholson method. A trinomial tree option pricing model can be shown to be a simplified application of the explicit finite difference method.

2.11 Model Formulation:

In finance, numerical procedures such as Binomial and trinomial lattice are used to determine the value of American and European options.

Black-Scholes Model Fisher Black and Myron Scholes [21] created a model for option pricing in 1973 which was formulated as a set of partial differential equations. This model revolutionized the option market and received Nobel Prize for Economics in 1997. This model can be used to find solution for European call and put options if values of five input parameters for options, then we can calculate the option value using this model.

The classical BSM formula for a call option is given

$$C(S, t) = N(d_1) \times S - N(d_2) \times K \times e^{-r(T-t)}$$

The Black-Scholes formula for put option is

$$P(S, t) = N(-d_2) \times K \times e^{-r(T-t)} - N(-d_1) \times S$$

“In these equations, S is the underlying asset price, K is the strike price in the contract, r is the interest rate, σ is volatility, t is the time and T is the maturity date. $N(d)$ represents the normal distribution function on d .”

$$Call = S_0 N(d_1) - Ke^{-rT} N(d_2) \text{ ----- Eq. 1}$$

$$d_1 = \frac{\ln(S_0/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}, d_2 = \frac{\ln(S_0/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}}$$

- S_0 is the resource price underlying
- K is the strike price for contract
- r is the interest rate
- σ is volatility (uncertainty)
- T is the maturity date of the option contract
- $N(d)$ is the probability that the option will be exercised

Figure: 2. Mapping Cloud Parameters to BSM

1. Input Cloud Parameters
 - Commodity Investment → Calculated through statistical analysis of particular resource
 - Option Contract Time
 - Resource Strike Price → based on resource depreciation rate (mostly assumed as 4 years [41])
 - Rate of Interest: For Amazon: 1.5% monthly or 19.5% yearly interest rate [42]
 - Volatility estimate from historical data (Amazon: 15%)
2. Map cloud parameters to BSM model to evaluate option price
 - S ← Commodity investment in cloud
 - K ← Strike price of resource
 - r ← Rate of interest
 - T ← Option contract time
 - σ ← Volatility estimate
3. Using BSM, calculate call and put using equations 3.1 & eq. 3.2 in pricing cloud resources

2.12 Cloud Federation

“A cloud model that, for the purpose of guaranteeing service quality, such as the performance and availability of each service, allows on-demand reassignment of resources and transfer of workload through a [sic] interworking of cloud systems of different cloud providers based on coordination of each consumers requirements for service quality with each providers SLA and use of standard interfaces”.

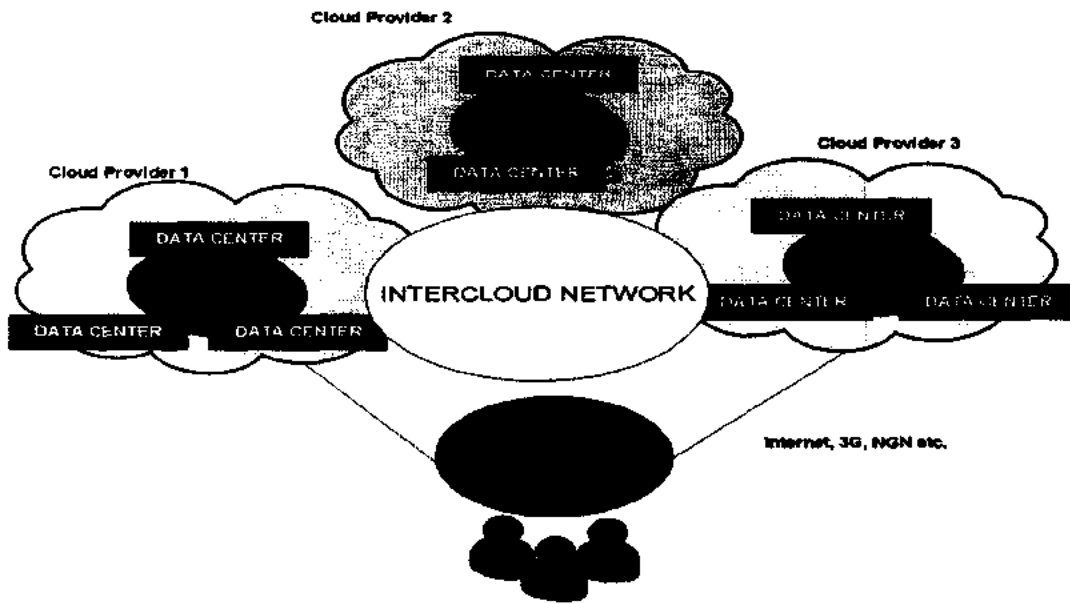


Fig 2.2 : Cloud Federation; Intercloud Network

Chapter 3

Chapter 3: Related Work

3.1 Overview

We are aiming to develop a new economic model to serve as the framework for pricing mechanism for cloud computing resources. In this regard, we have conducted some extensive review of existing literature about cloud computing pricing models. We have learnt from study of literature that there is no standard pricing mechanism and there is no generalized framework for this purpose. We considered the use of financial Option theory for establishing the pricing model for cloud computing resources.

3.2 Pricing of the Cloud

In “Present or Future: Optimal Pricing for Spot Instances” [9] , Wang et al investigate the problem that how cloud provider sets the spot price to maximize its revenue. First, they present a demand curve model which precisely captures the attributes of spot resources for current cloud computing. This model also takes the impact of pricing on the future into consideration. Further develop the time-average revenue maximization problem for cloud providers and present an algorithm to solve the time-average revenue problem. The algorithm applies Lyapunov optimization which operates with and without any knowledge of the future requests. They have illustrated the motivation of their work using example of spot instances of Amazon Elastic Compute Cloud (EC2).

Source	Statement on pricing
Armbrust et al., 2009	“...made available in a pay-as-you-go manner to the general public...”
Buyin et al., 2009	“Consumers are then able to pay service providers based on their usage of these utility services”
Durkee, 2010	“The essential characteristics of cloud computing that address these needs are: ... Pay-per-use . Much like a utility, cloud resource charges are based on the quantity used.”
Foster et al., 2008	“In a cloud-based business model, a customer will pay the provider on a consumption basis , ... such as electricity, gas, and water ...”
Gong et al., 2010	“... when a user use the storage service of cloud computing, he just pay the consuming part without buying any disks ...”
Grossman, 2009	“Cloud computing is usually offered with a usage-based model in which you pay for just the cloud resources that a particular computation requires ”

The authors investigate dynamic pricing problem for spot instances to maximize the expected revenue of provider for long time. They propose a arrival-departure model to characterize the impact of current price on the future demand. Then, a dynamic programming based approach

is proposed to solve the dynamic pricing problem. However, this approach requires the statistics of request arrival which is usually hard to obtain in practice.

According to [9] they considered the case of a single cloud provider and addressed the question how to best match customer demand in terms of both supply and price in order to maximize the providers revenue and customer satisfactions while minimizing energy cost. That problem is modelled as a constrained discrete-time optimal control problem and by using Model Predictive Control (MPC) to find its solution. Simulation studies using real cloud workloads indicate that under dynamic workload conditions, the proposed solution achieves higher net income than static allocation strategies and minimizes the average request waiting time. But that research was only considered CSP's perspective and not fulfilled the customer's significance.

Keeping in view the research available in literature, we intend to develop a model for Spot VM for better resource allocation. And also for the customer's concern of optimal bidding so minimum cost can be achieved.

Author's of [11] has tried to develop a Flexible accounting model for cloud computing, classic solutions fail to provide a proper answer as they were not specifically design for cloud computing, their proposed accounting model that allows the deployment of cloud computing services to accomplish all the service providers' requirements and interests.

One important part of the accounting process are the Pricing schemes. And there are multiple charging schemes may use different types of services e.g. Time based, Volume Based, QoS based, flat rate, service type, free of charge, discounts, content based, location based, usage based, smart pricing, edge pricing, priority pricing, dynamic pricing, static pricing.

In the case of cloud computing, due to its different nature, the most used pricing schemes so far are time-based and utility-based pricing (charge by provision). Time-based schemes in cloud computing pricing varies from service to service but the business formula is always the same: multiplying a fix price by a consumption time. In the case of utility-based pricing in cloud computing, the scheme may also vary from service to service.

But is not the same to model a *Software as a Service (SaaS)* provider or a *Infrastructure as a Service (IaaS)* provider. According to time-based , SaaS the utility is the service itself and consumption time is normally measured in number of uses. As for IaaS, time is measured in hours.

Utility-based scheme charges the user on a per-use basis and its complexity relies in controlling the operating costs. These models have been working properly in this early state of cloud computing. Business requirements, however, are changing and the introduction of other pricing schemes is essential. In order to enable this process in a flexible way, new engineering efforts are required.

According to authors of [18] they developed an online algorithm **OPT-ORS**, by using Lyapunov optimization framework to explore the trade off between the procurement cost and the user's Quality of experience QoE for cloud based video streaming. They have formulated the problem as a joint optimization problem of resource provisioning and procurement price. They use all three types of VM's offered by Amazon EC2 i.e. Reserved. On Demand and Spot. Video service providers can use multiple pricing models to optimally procure the number of VM instances to satisfy dynamic user demands. And it resulted as spot instances are rent more frequently than both other instances to serve user demand due to their low cost, but no specific bidding strategy is assumed which should be considered. Because optimal bidding is very important for provisioning of Spot VM, and Price is not the only factor to be measured. There are other important factors which are critical for completion of a task in Cloud Computing.

On the side of resource allocation, and Buyya and Voorsluys [31] solve the problem of work-intensive calculation running on a set of intermittent VMs. To mitigate possible periods of unavailability, the study proposed a multifaceted political mistake aware of them provide resources. Their solution employs pricing mechanisms and assesses the runtime. The proposed strategy and achieve cost savings and strict compliance with deadlines.

Zhang et al. [12] presented a solution of the best ways to adapt to customer needs in terms of supply and price for service provider maximize customer satisfaction in terms of scheduling VM. The proposed model has been designed to solve the optimal control problem discrete time. This model achieves greater distribution strategies of fixed income and reduces the average waiting time demand. Our work differs from [7,8] that we focus on reducing the

decrease of time after the failure of the mission, and to achieve cost savings and reduce the total runtime

3.3 Financial Option theory in Literature:

The idea of Financial Options is used by Allenator et al. [14] to price the grid compute commodities (gcc). They designed a Fuzzy Real Option Model for finding the value of grid resources i.e. gcc. Gcc was treated as real assets because of nature of gcc being transient i.e., flexibility opportunity and is valued through fuzzy logic framework in a discrete time approach. The trinomial model/lattice is used to solve the real option pricing problem. The Grid Fuzzy Real Option Model (G-FRoM) consists of four levels: (i) the resource modeling, (ii) monitoring and notifications, (iii) accounting and auditing, and (iv) user application based functions. G-FRoM provides a controlled system between the price and utilization of grid resources, which can guarantee profit against optimal resource utilization as well as assurance of user's satisfaction. The model is evaluated with usage of real data provided from WestGrid Canada. It is considered to be the first study which used financial options as a pricing mechanism for grid resources. It also takes into account the effects of technological aspects.

The above study was further enhanced by the same authors [15] to provide a balance between Quality of Service (user's perspective) and Profit (Service provider's perspective). The trinomial model, fuzzy logic and Price variant factor were employed. The equilibrium between QoS-Profit was achieved by introducing Price Variant Factor which is a fuzzy number $[0,1]$. The study was conducted upon simulation of data usage from two real grid nodes. American put and call options were applied. The option values were obtained and studied to find out the variation in space of six months.

In [16], Allenator et al. considered data usage of three grids of different nature i.e. SHARCNET, Grid5000 and Grid-3. Grid resources were priced using financial option theory by executing one step trinomial lattice using the parameters of strike price, resource price, expiration time, interest time, volatility and number of steps. The option prices were obtained from simulation. The results pointed out that cost of grid compute commodities is time depended so it is hard to predict the exact price of grid compute commodities in actual life. Through this model, the best time to exercise the option can be determined.

A further study was conducted by integrating the financial option based pricing architecture onto the top layer (Price Usage and optimization Level) of GridSim toolkit [17]. The option price was calculated by running the trinomial lattice. The price and usage optimization level is developed by optimized usage of resources and option price and this layer was integrated on the GridSim user codes. Dutch auction was used to reduce the cost of resources so maximum usage of resources can be achieved. The financial option based pricing architecture is justified with scenarios developed in GridSim.

The simulation data from the same pricing model integrated with GridSim was evaluated against real data from six different grids [18]. The results were analyzed and it was concluded that resources availability do not remain constant at all times, for example few grids can provide resources to user at high price and sometimes unable to do that. In [19], the same behavior was observed by using two specially selected grids, one was LCG which was an experimental platform grid and other was a commercial grid i.e. Auvergrid.

Financial option theory was proposed for cloud federation [20] to overcome the problem of underutilization of resources and QoS for reserved users (availability of resources when they require the resources). The problem was addressed by trading (buying or outsourcing) the required resources from any other service provider in the cloud federation. It was induced from the experimental results that the use of option contracts for resource reservation can be helpful in future, when service provider will need additional resources, there won't be any need to buy these resources from other service provider in the cloud federation. It will increase provider's profit and user's trust in that provider because there won't be any chance of service denial or unavailability. One limitation of the study was that selling (put) options were not considered.

A Cloud Asset Pricing Tree (CAPT [21]) was designed, simulated and evaluated to find out the optimal premium price for Cloud Federated Options. It provided the benefit to the service Provider in terms of decision making i.e., when to buy options in advance and when would the best *Time* to exercise them, so that maximum saving can be made in provisioning the Virtual Machines. Financial Option theory is employed as an interface/facility to share an extra pool of federated resources whenever needed.

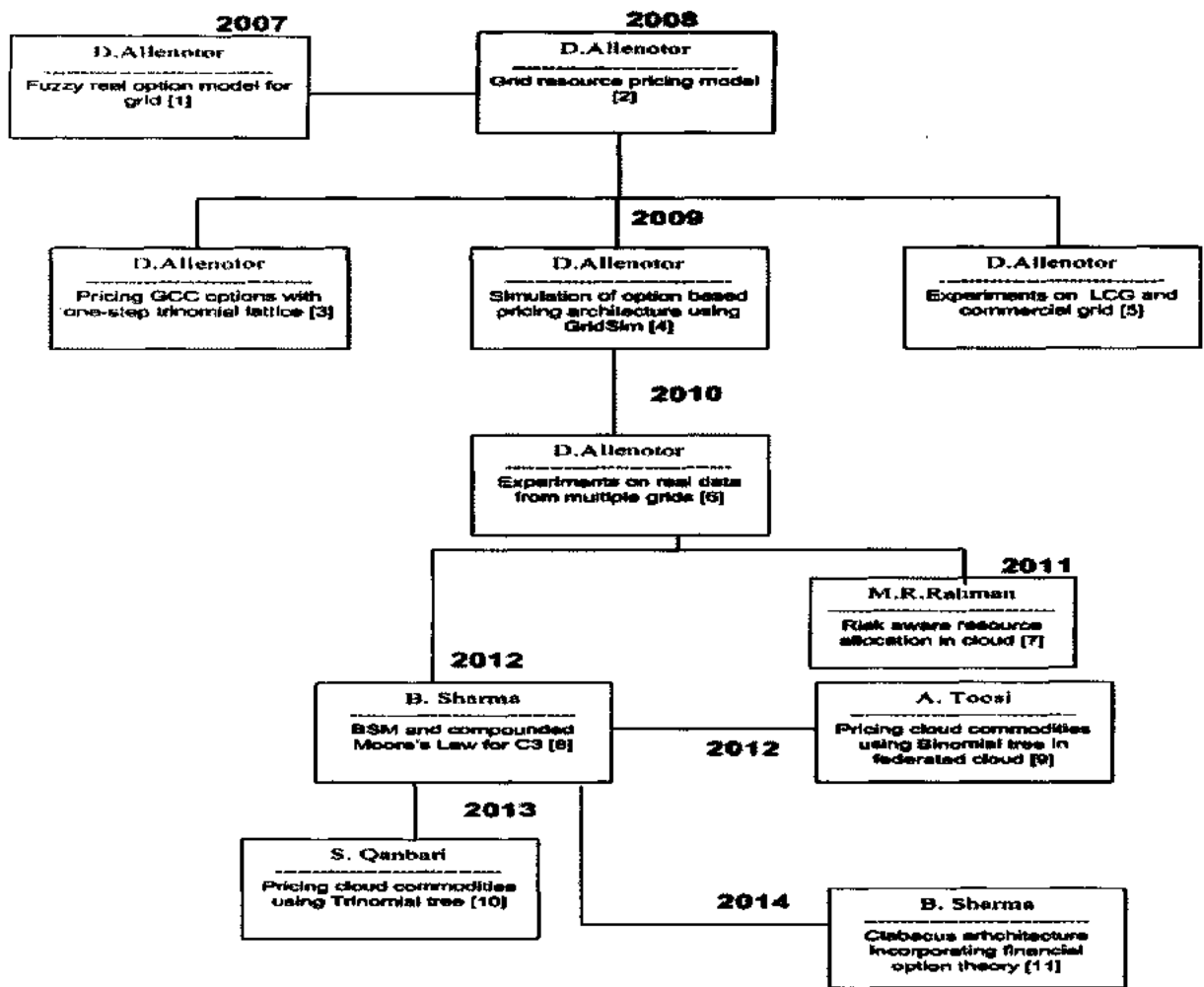


Figure 3.1: Pictorial Representation of Financial Option Theory

From the literature mentioned above, it is evident that financial Option theory is suitable for Grid Computing. However, a useful theoretical study for cloud computing pricing has been introduced by Sharma *et al.*, [22]. They proposed a novel financial economic model which is able to provide a high level of Quality of Service to consumers. Financial option theory was used and considered the cloud resources as assets to capture their real price. The price determined using this model represented the best possible price that the service provider is supposed to charge its customers to recover the primary/capital costs. A lower boundary on the price that should be charged to customers is given by the financial option theory. The upper boundary of the price was calculated using a proposed compounded Moore's law. Moore's law was combined with the compounded interest formula to form a compounded law. The authors stated that, if the price should be set between these two boundaries, which could be useful for customers and service providers equally.

The authors of [23] have proposed an approach that utilizes financial option theory to alleviate risk and reduce cost for cloud users in spot markets at the same time. The cloud user optimization problem was formulated and mathematically characterized the cost of using European style options for clouds. A novel on-line policy using American options was proposed that overtakes standard spot policies in terms of price variance reduction against high risk factors.

Chapter 4

Problem Definition

4.1 Introduction

Cloud computing has grasped the attention of scientific community and business industry towards the provisioning of computing resources as utility and software as a service over a network. Gartner Forecasts worldwide businesses and individuals spending on Cloud Computing services is expected to be \$112 billion in 2017. Profitability and revenue maximization are the most important goals for any cloud service provider, which can be employed through different pricing models; however, end-users are typically more interested in high satisfaction guarantee through Quality of Service (QoS), cost-effectiveness, usability and availability of cloud resources (Users maximize utility and CSPs maximize profits). Keeping a balance between these two (trade-off) is the most challenging design decision to be made by cloud service providers.

In cloud, provisioning of computing resources is offered in the form of Virtual Machines (VM), being deployed on physical computing nodes/ servers. Cloud data center needs to be efficient and scalable to connect thousands and even thousands of thousands of such physical machines. However, installation, configuration and management of these hardware resources poses an important problem: Time-varying patterns of cloud load over different data centers.

The another problem is of finding the optimal bidding mechanism for successful acquisition of VM i.e., which price to be offered which could help in providing the low cost to customer and when the bid increases the current offered price, how to avoid the risk off job termination. Because whenever the bid price increases the currently running jobs on Spot instances are stop, customer has to re-bid on higher than previous price.

There are various types of VM instances offered by cloud service provider at different prices, with different compute capabilities ,network and storage. Amazon Elastic compute cloud provides three pricing models, On demand, reserved and Spot instances.

In case of on demand instances customer is charged for VM resources on the basis of compute capacity, and the issue is there is no long term commitment, if the cloud provider suffers from over provisioning of resources , the instance can be interrupted/stop at any time without intimation.

Reserved instances require one time collective payment in advance for the long term use of resources. Whereas in spot instances , customer can bid for unused Amazon EC2 instances. Spot price is determined by supply and demand , if the bid is higher then the spot price, instance is get. But as soon the bid price falls below the spot price, instance is interrupted.

4.2 Problem Statement

Current payment system for cloud resources is flat i.e. pay as you go. Objective of this research is to propose a dynamic payment model to optimize over all resource utilization for cloud provider and .

- Pricing is based on the type and size of instances, required resources and sometimes it considers the region as well. The traditional business models are fixed cost model.
- In cloud computing, no cost model has been developed yet that builds a distinction between variable and fixed cost and specially for Optimal Bidding of Spot VMs
- Efficient resource provisioning model for the unused resources to maximize the cloud provider's profit which could be useful instead of facing the problem of underutilization of resources.

Chapter 5

5.1 Proposed Framework:

In virtual cloud, cloud provider, underlining public cloud infrastructure, and cloud service provider realizing cloud resources/ services may be different vendors. Resources of other cloud providers are normally borrowed to meet end user requirements. Cloud service provider does not itself own networking or data center resources. A cloud consumer can construct virtual cloud by leasing virtual machines from the cloud providers. A central entity (also known as broker or mediator, global cloud agent/coordinator) performs or facilitates multiple clouds to share resources. Cloud broker acts as an intermediary between service consumers and producers. Cloud consumers can find best provider and service through the matchmaking process of cloud broker.

The primary design goal of our proposed system is to facilitate user job execution by automating the entire process on hand and achieve economic efficiency by exploiting low-cost spot VM on the other hand. The core components of the system are cloud customizer, broker at user end and server side cloud provider component.

A cloud broker may also provide customers with some additional services, encryption and transfer of consumer data to the cloud and monitoring data life cycle management. Such broker is known as cloud enabler or cloud aggregator. Sometimes a broker integrates cloud services on behalf of customers to work together and sells the services under their own brand; such broker is known as cloud customizer or white label cloud service (source: <http://searchcloudprovider.techtarget.com/definition/cloud-broker>).

Our proposed cloud customizer involves the following steps:

- Job Admission: Jobs are submitted by the users along with necessary information including task(s) to be executed, budget, deadline etc.

- **Runtime Estimation phase:** When a job is submitted, the broker estimates the job characteristics and schedule amount of processing nodes considering the workload requirements. Our Parallelism profile component uses Downey's model to extract such features. More details about the model is discussed in Section II.
- **Discovery phase:** Cloud customizer queries a list of resource/ service providers that satisfies the requirements.
- **Resource Selection:** Checks individual resource / service provider to confirm the service requirements. The cost of executing a task is obtained by querying cloud provider. A final priority order list is generated after confirming each individual provider. Spot instances are given the highest priority for execution of non-critical jobs.
- **Scheduling Module:** This upper layer scheduling module is responsible for creation of the virtual machine pool according to actual state information of the user job. Based on the priority order, the module aims to complete job within budget and deadline constraints. More details about the scheduling module is discussed in section IV.
- **Resource monitoring:** This module of broker continuously monitors secured resources against service abruption, violation of SLA, QoS etc.
- **Resource switchover:** In case of early termination in case of Spot VM or low QoS in case of on-demand instances, provisioning of resources is re-evaluated to meet QoS and deadline constraints
- **Release of Resources:** Unused resources are released after successful execution of job

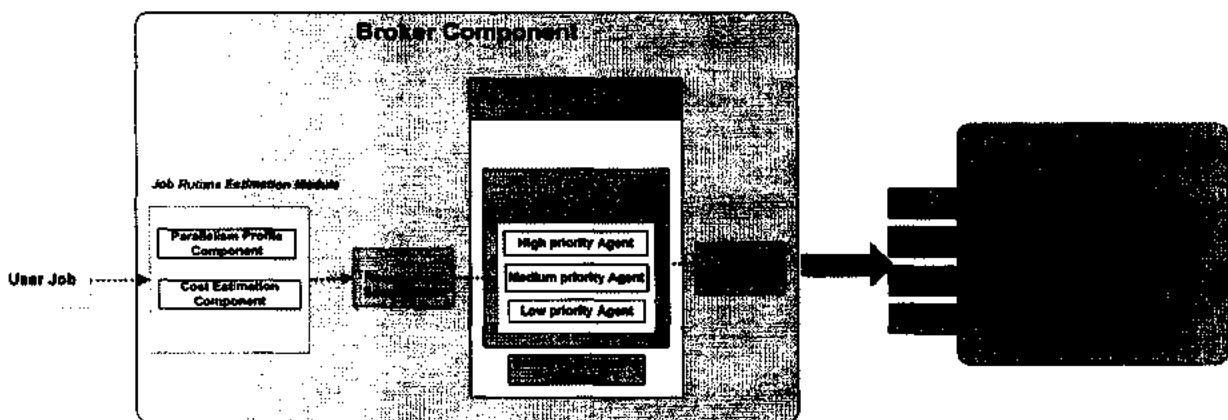


Figure 5.1: Overall process workflow of broker and cloud service provider components

Our proposed model uses spot instances to utilize the economic benefits of low cost computation model of spot instances. However, for high priority and deadline constrained jobs, financial options are exercised to guarantee the execution of critical jobs. If job execution in spot instance is failed, the job is re-checked by our provisioning algorithm to place it either in spot instance queue or on-demand option queue according to its priority. Basic flow of events is given as under: High priority jobs are critical and deadline constrained hence on-demand instances from cloud federation using financial options (future contracts). Such instances are pre-reserved by paying a premium price in advance to meet the workload requirements in peak hours. The primary benefit of this model is to achieve more economies of scale on as well as avoid job rejection during peaks.

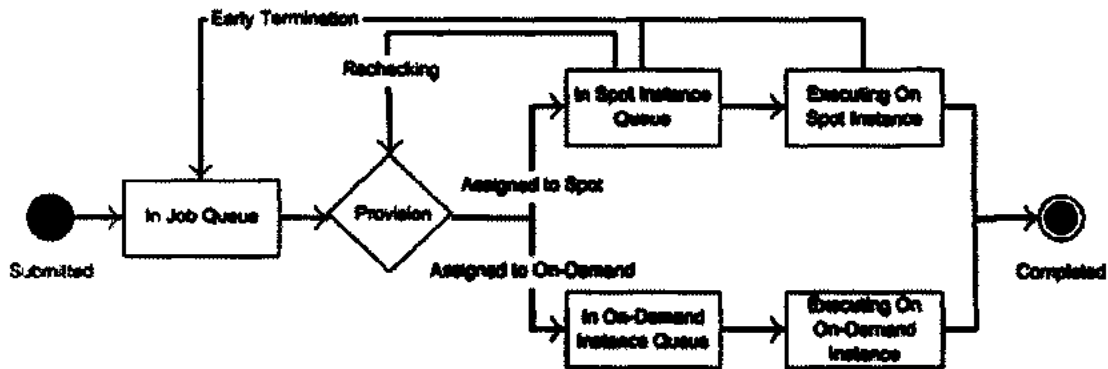


Figure 5.2 : User job lifecycle

5.2 Job Runtime Estimation Module

Parallel workload management and job scheduling is an important area of research in distributed computing. In traditional super-computing model, parallel job execution is achieved by in-house application development. However, in case of cloud computing, users are mostly unaware of parallelism and they don't even know the amount of processors required to execute their job within time and budget constraints.

In cloud computing environment, mouldable job scheduling (processing units are determined considering the free nodes) is more feasible to reduce the average turnaround time of execution. Resources can be allocated at either submit time or schedule time. User feedback, if required, can also be incorporated to calculate maximum allowable amount of processing

nodes. Kuo-Chan Huang et al. [] argued that schedule time allocation outperforms submit time considering the complex infrastructure requirements of cloud computing.

We use Downey's model of speedup [23] to estimate the job characteristics and schedule amount of processing nodes. The same model was also demonstrated by Voorsluys et al. [] for job speed up. If L is the sequential runtime of the job and $T(n)$ is the execution time of the job on n processors, then the speedup function $S(n)$ can be calculated as: $S(n) = L/T(n)$. The model is a non-linear function of two parameters: A denotes the average parallelism of a job and is a measure of the maximum speedup that the job can achieve; σ (sigma) is an approximation of the coefficient of variance in parallelism within a job. Higher the value of sigma, higher is the deviation from linear case. Parallelism profile is constructed using parallelism of the program, A , and the variance in parallelism, V . Two types of profiles, one for programs with low variance, the other for programs with high variance, are constructed using this model.

For low variance ($\sigma \leq 1$)

In this case, the parallelism is equal to A , the average parallelism, for all but some fraction σ of the duration ($0 \leq \sigma \leq 1$). The variance of parallelism is $V = \sigma(A-1)^2$.

Following runtime as a function of number of processing units, n , is defined:

$$T(n) = \begin{cases} \frac{A-\sigma/2}{n} + \sigma/2 & 1 \leq n \leq A \\ \frac{\sigma(A-1/2)}{n} + 1 - \sigma/2 & A \leq n \leq 2A-1 \\ 1 & n \geq 2A-1 \end{cases}$$

Based on the equation given above, $T(1) = A$ and $T(\infty) = 1$.

Speedup can be defined as:

$$S(n) = \begin{cases} \frac{An}{A-\sigma/2(n-1)} & 1 \leq n \leq A \\ \frac{An}{\sigma(A-1/2)+n(1-\sigma/2)} & A \leq n \leq 2A-1 \\ A & n \geq 2A-1 \end{cases}$$

For High Variance ($\sigma \geq 1$)

Here, σ is approximately square of the coefficient of variation of parallelism, CV^2 which is equivalent to $\sigma(A-1)^2/A^2$.

Following runtime as a function of number of processing units, n , is defined:

$$T(n) = \begin{cases} \sigma + \frac{A + A\sigma - \sigma}{n} & 1 \leq n \leq A + A\sigma - \sigma \\ \sigma + 1 & n \geq A + A\sigma - \sigma \end{cases}$$

Thus $T(1) = A(\sigma + 1)$ and $T(\infty) = \sigma + 1$.

Speedup is given as under:

$$S(n) = \begin{cases} \frac{nA(\sigma + 2)}{\sigma(n + A - 1) + A} & 1 \leq n \leq A + A\sigma - \sigma \\ \frac{A}{A} & n \geq A + A\sigma - \sigma \end{cases}$$

Based on the above model, the scheduler would be able to select required number of resources considering the workload requirements.

5.3 Scheduling Policy

The scheduling module assigns task to the pool of virtual clusters according to the job specifications. It is responsible for completion of job execution within the budget and deadline. Since jobs are executed based on priority order list, we introduce the priority level (PL), given as equation I, the maximum estimated time for job in a wait queue before the deadline is reached. If PL is negative, job deadline cannot be met. The greater the value of PL, the more is the chance to meet the deadline and hence job can be placed in the low priority. A small positive number indicates the resource must be provisioned immediately and job is considered to be in high priority.

$$PL = \max(0, T_{deadline} - T_{now} - (a * T_{est} + T_{latency})) \text{ -----} J$$

where $T_{deadline}$ is the job's deadline specified by user, T_{now} is the current time, T_{est} is the estimated time of job as calculated by the job runtime estimation module, $T_{latency}$ is the expected time to set up VM instances. a modifier is the sensitive factor []; higher values of modifier indicates the scheduler to provision on-demand instances as PL tends to zero. Low values of modifier places the job in a low priority queue to exploit the economic benefits by utilizing spot instances. To ensure that jobs with low priority eventually executes, we use the concept of aging. Priority of low level job is gradually increased (upon every failure/interruption) and hence a low level job may get high priority to complete its execution.

All incoming jobs are received by the job manager agent J . Considering the job priority list, the job is enqueued in one of three job queue agents: JA_{hi} for high priority jobs, JA_{med} for medium priority jobs and finally JA_{low} for low priority jobs. For high priority jobs, priority level is set to some positive number below one to indicate that job can only be run once.

Medium and low priority jobs are executed on spot instances as spot instances are offered on very cheap prices (as low as 1/3 of on-demand instances). This scheme enables cost-aware task scheduling for budget constrained jobs. Pricing bidding strategy is based on Amazon spot price history. JA_{low} accepts jobs below a specific threshold τ which is set by the user. If $\tau=3$, user job can run at least twice on spot VM. Keeping track on the average spot pricing history of past 90 days, the bidding strategy for JA_{med} is relatively aggressive (above average price) as compared to JA_{low} so that the chance of spot termination due to out of bid would be relatively low. JA_{low} accepts all jobs above τ . Since the job has multiple chances for execution, JA_{low} bids on relatively low price (around average price) considering the spot history. If a task is interrupted, it is sent back to job pool J to reschedule according to its priority level.

5.4 Algorithm for VM provisioning and job scheduling

```

While (true) do
while current time < next_schedule_time    do
queue incoming job j in J
vms ← all VMs currently in pool
for each  j in J

```

```

compute priority level (PL)
if j is JAhi then
    exercise options
    continue;
else
    if (j is JAme or JAlo) then
        P ← find average spot price history /* apply necessary bidding
        strategy */
        if (j is JAme) then
            j.bid=P+G
        end if
        if (j is JAlow) then
            if j.bid ≤ P
                j.bid=P
            end if
        end if
        end if
        mwt ← maximum wait time for j
        decision ← FindFreeSpace(j.bid, vms)
    end if
    if (decision.allocated=true) then
        AllocateJobToVM(j, decision.VM)
    else
        add j to list J
        delay allocate time by mwt ratio
    end if
end for each

```

Priority Level
$PL = \max(0, T_{deadline} - T_{now} - (\alpha * T_{est} + T_{latency}))$
Greater α , more sensitive algorithm becomes. $\alpha = 1$

JOB CLASSIFICATION
High, Medium, Low
JA _h JA _M JA _L

Chapter 6

Chapter 6

Implementation

6.1 Simulation Experimental Setup

We have implemented the proposed framework through simulation. For this purpose we used java based simulator “CloudSim” which is developed by CLOUDLAB University of Melbourne. In this research, we have used following tools to carry out the process of simulation. **CloudSim 3.0, CloudReports 1.1.**

CloudSim is a simulation framework used for modelling and experimental simulation of Cloud strategies. It can be further extended by researchers to carry out their specific experiments to test the performance of different cloud computing scenarios. It is a java based tool, which consists of multiple classes. It provides console based results to perform synthesis, so we have also used CloudReports that is an extension of CloudSim which provides GUI based statistical synthesis of results. CloudReports is wrapped around CloudSim.

6.2 Basic Concepts in CloudSim

1. Datacenter

Datacenter is composed of a set of hosts and is responsible for managing virtual machines (e.g., VM provisioning). It behaves like an IaaS provider by receiving requests for VMs from brokers and creating the VMs in hosts.

2. DatacenterBroker

This class represents a broker acting on behalf of a user. It modifies two mechanisms: the mechanism for submitting VM provisioning requests to datacenters and the mechanism for

submitting the tasks to VMs. We had extended this class for conducting experiments for our designed policies. Since the default behaviour of broker in Cloudsim is round robin scheduling, we have extended the broker class to match our scenario and provision resource based on four different policies, discussed in sub-subsequent sections.

3. Host

Host performs tasks related to supervision of VMs which includes creation and destruction of VMs and updates task processing to VMs. A host has a clear policy for provisioning memory, processing elements, and bandwidth to virtual machines and is associated to a datacenter. A set of multiple virtual machines can be hosted by a single host.

4. VM

VM is depiction of a software implementation of a machine that runs applications called virtual machine which works like a physical machine. Every single virtual machine splits the resources received from the host between tasks running on it.

5. Cloudlet

The tasks are implemented in the cloudlet class and are termed as cloudlets. The complexity of an application represented by CloudSim in terms of its computational requirements. DatacenterBroker Class is responsible for the implementation of scheduling policy which manages the requirements.

Basic procedure of CloudSim includes the creation of Datacenters, Hosts are created in these data centers while VMs are created through Hosts. DatacenterBroker overall manages the creation of hosts and VMs in these hosts. Customers issue requests for creation of VMs, the tasks are further divided into cloudlets. These tasks are allocated to different VMs.

We have extended the DatacenterBroker class to implement our policies i.e., On-Demand resource provisioning scheme, Spot VM resource provisioning scheme, Resource provisioning scheme with feedback and Runtime estimated resource provision scheme (without feedback). The extended Broker Class performs the main task of job scheduling according to our designed policies whereas, by default, the policy used is Round Robin. In Round Robin, the VMs are allocated randomly, the workload is divided among available hosts in Datacenters. We have implemented suitable extensions, modifying the java code in

the extended broker class, which divides the workload according to user requirements, providing low cost and optimal resource allocation. The cost model was extended as to accumulate the per hour processing cost of on-demand and spot prices as CloudSim does not calculate the cost according to per hour VM specifications.

The other extension, we have applied in CloudSim code, was setting different priorities of job as required by users in our case. Keeping in view the previous examples of jobs failure, huge losses can be avoided by migrating/switching the critical and deadline constrained jobs to other suitable datacenters instead of waiting for availability of Spot instances. We have extended the VMAllocationPolicy which performs VM allocation to different data centers instead of single data center is populated with burden of all tasks. Dividing the workload between different data centers lowers the spot price too as more resources becomes free.

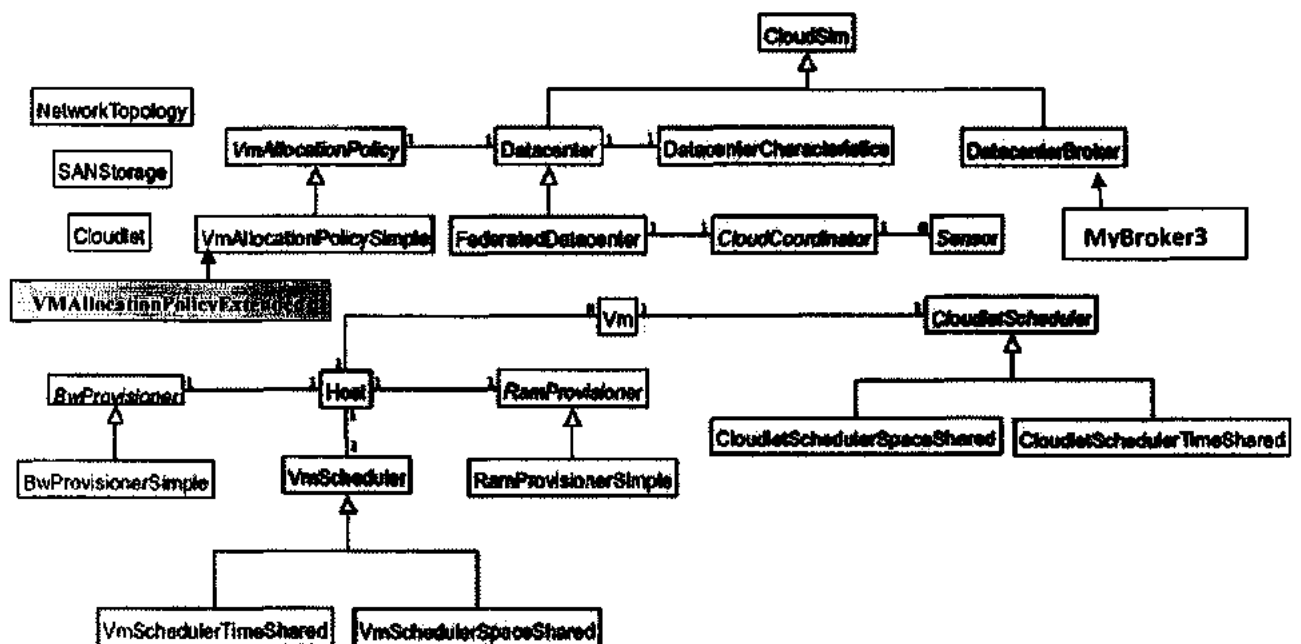


Figure 6.1: Architecture of CloudSim Extended

In our experimental setup, we have tested our four policies by executing simulations multiple times in CloudReports. We performed a set of experiments by creating 10 to 140 VMs by an increment of 10. During the experiments we measured following variables:

1. Cost
2. Processing load
3. Time

6.3 Option Pricing

For to hedge against the unavailability of computational resources for deadline constrained job(s), we used financial option based leasing strategy. In such scenario, resources are leased for time T by paying a premium price known as call options. In our case , for resource provisioning policies (3) and (4) we used options with the following configuration:

The screenshot shows a software interface for options calculation, divided into two main sections.

Underlying Type:

- Underlying Type: Futures
- Asset Price: 0.07
- Volatility (% per year): 15.00%
- Risk-Free Rate (% per year): 7.00%
- Buttons: Calculate, Display Tree

Option Type:

- Option Type: Binomial: American
- Implied Volatility:
- Time to Expiration: 0.0833
- Exercise Price: 0.07
- Tree Steps: 10
- Option Style: Put, Call

Figure 6.2: Provided Parameters for Options Calculation

Based on these parameters, the following binomial tree was generated and used in our experiment. Since the option values with ZERO premium price are not realistic , we used option values ranging from 0.0019 to 0.0102 depending on supply and demand of VM resources.

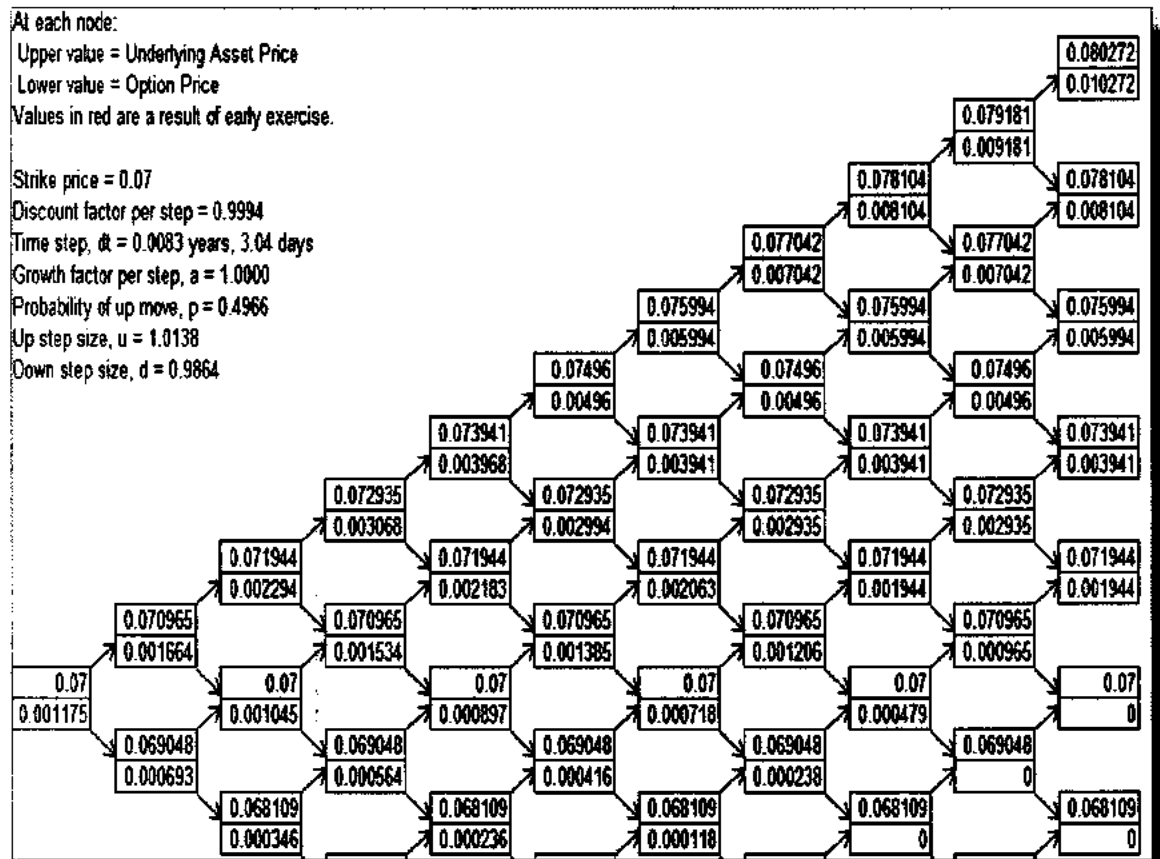


Figure 6.3: Binomial Tree of Options

We used price history for estimating the price and Spot VM workload from the following figure.

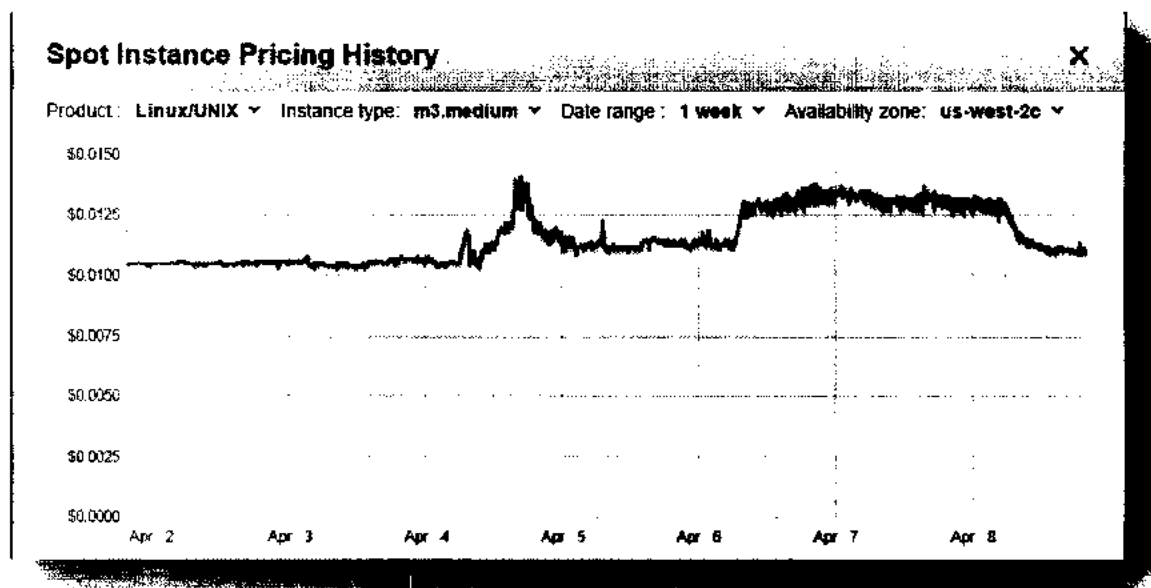


Figure 6.4:- Amazon price history for spot instances of one week

Source:<https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#SpotInstances:sort=requestId>

Based on the Amazon spot price history, we generated spot VM workload where the peaks represent high spot VM bidding competition while the valleys represent spare capacity that can be utilized without market competition.

6.4 Results and Discussion

We used Amazon pricing data for our experiments. For most of our experiments, we calculated per hour cost, based on Amazon m3 medium instances:

VM Type	On Demand Cost (\$)	Spot VM Cost (\$)
m3.medium	0.07 per hour Additional Cost: 0.001 per GB storage and network operations	0.01 to 0.015 per hour Additional Cost: None

Table 6.1

To evaluate an optimal policy based on deadline constrained, budget constrained and cost effective optimal application centric resource provision scheme, we derived four different policies: (1) on demand resource provision scheme (2) Spot VM resource provision scheme (3) resource provisioning scheme with feedback (4) runtime estimated resource provisioning scheme (without feedback)

6.4.1 On Demand Resource Provision Scheme

Under such scheme we used a pool of on demand instances to complete job requirements. In our simulation environment these resources were released for a series of experiment ranging from 10-140 VM requests. The simulation results based on CloudSim reports somehow show a linear relationship between number of VMs and overall cost incurred. The results are show in fig.

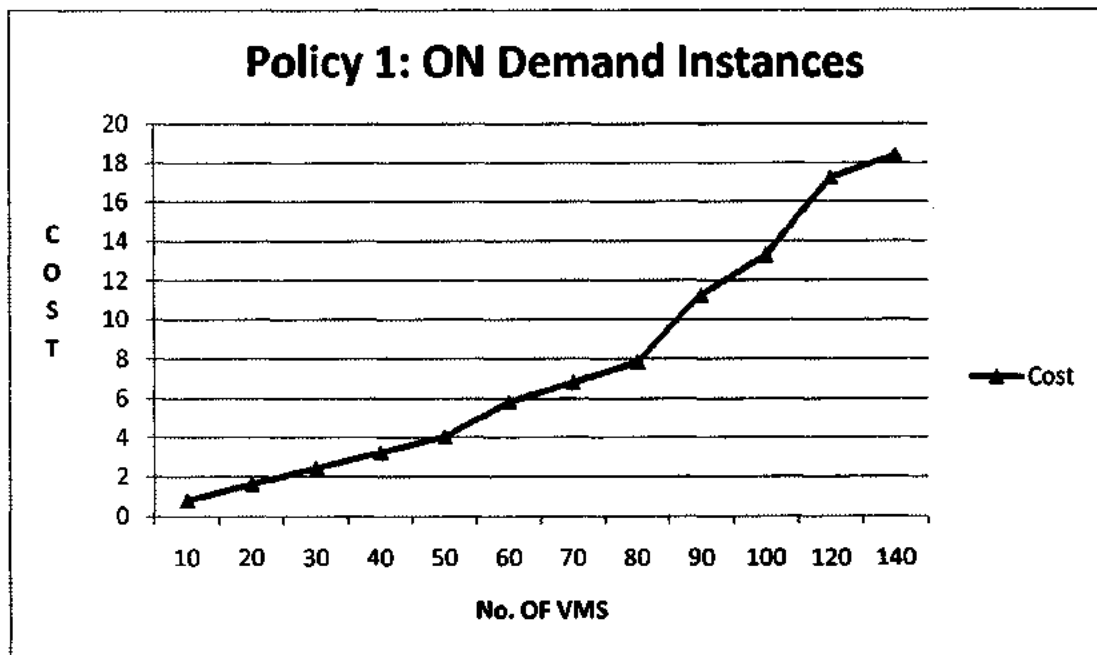


Figure 6.5: Policy 1 On-Demand Instances Resource provision Scheme

Although the job requirements were successfully met, overall cost was relatively higher and such policy can only be implemented for deadline constraint jobs which is not a very frequent case in cloud computing environment. One important point that can be observed in figure above is the significant increase in price between the VM range 80 to 140. This is quite understandable as with the increase of VMs, the bandwidth and other operations tend to rise and hence as a result overall cost increases.

On Demand cost	No. of VMs
0.812	10
1.617	20
2.429	30
3.234	40
4.046	50
5.824	60
6.797	70
7.847	80
11.249	90
13.272	100
17.234	120
18.368	140

Table 6.1 : On Demand

6.4.2 Spot VM Resource Provision Scheme

In this policy, all VM requests were executed on Spot instances. In case of spot VM termination due to relatively low bid, the job was rescheduled unless deadline schedule is not over. Since VM requests were randomly generated, some VMs submitted in the mid of simulation were not executed successfully. Furthermore, with the increase of Spot VM demand, the rejection rate was also more frequent as compared to low workload. We used The same workload model of Spot VM instances as presented in figure Amazon price history graph for one week.

Spot Price	No. of VMs	Rejection Rate
0.12	10	0
0.491	40	2
0.87	70	13
1.025	80	16
1.47	100	24

Table 6.3: Spot Prices

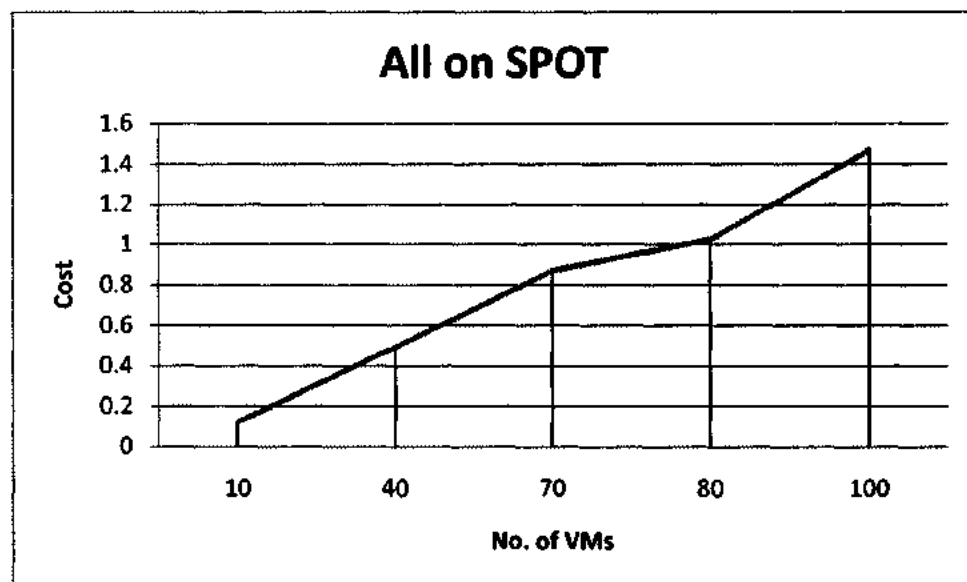


Figure 6.6 : Plot of Cost-No. of VMs for All VMs on Spot

Although the results are promising in terms of financial cost and many companies have cut down 50%-60% of their expenses using Spot VM instances, this scheme is not preferable for interactive and real time job allocation. A famous example is SEOMOZ's "Crawler" where all spot VMs were terminated without any prior notice and SEOMOZ suffered huge financial loss. As a lesson learned, the company had to decide a mix strategy consisting of both On Demand and Spot VM to achieve better resource provisioning as well as minimizing overall cost.

Hence in this research, we introduced two new strategies to achieve resource utilization efficiency, meeting budget and deadline constraints and while keeping the cost low. These two policies are discussed in subsequent sections.

6.4.3 Resource Provisioning Scheme with Feedback

In this scheme, an incoming job request can be classified into low, medium or high priority. According to the algorithm discussed in detail in section 5.3, following process takes place:

Based on the job classification all incoming jobs are enqueued in one of three job queues agents: JA_{hi} for high priority jobs, JA_{med} for medium priority jobs and finally JA_{low} for low priority jobs. For high priority jobs, priority level is set to some positive number below one to indicate that job can only be run once. For such jobs the risk of unavailability of VM resources may result in job failure. To hedge against this risk, we use the concept of advanced reservation through financial options. For such jobs, options are purchased by paying a premium price and may be exercised, when required. This strategy minimizes the risk of job failure. Option price calculation is discussed in section 5.2 in more detail. Medium and low priority jobs are executed on spot instances. This scheme enables cost-aware task scheduling for budget constrained jobs. Pricing bidding strategy is based on Amazon spot price history. If a task is interrupted, it is sent back to job pool J to reschedule according to its priority level. Following figure shows 100 VM job request with different priorities assigned

by

user:

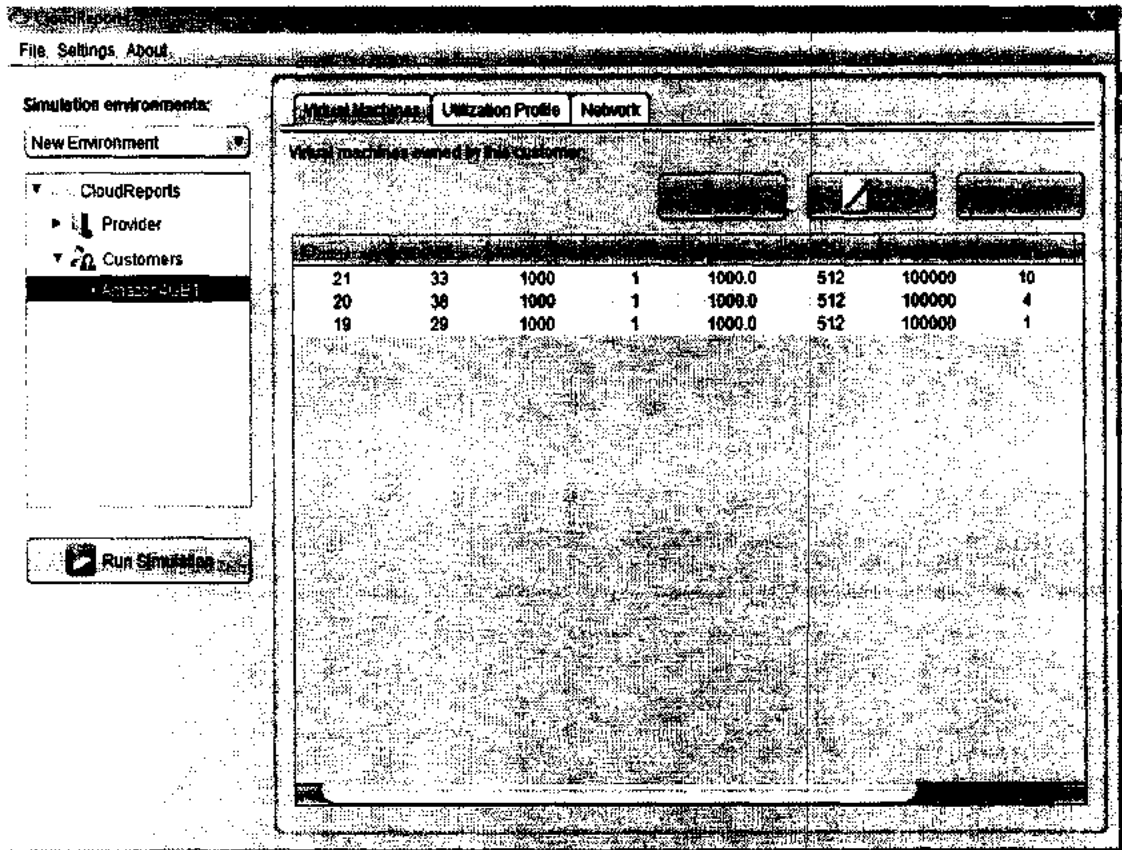


Figure 6.7 : Simulation Setup of 100 VM request with user feedback.

The experiment results are presented as follows:

No. of VMs				
10	0	6	4	0.40
40	3	21	19	1.79
80	21	27	53	5.54
100	30	30	70	9.73
140	45	39	101	13.83

Table: 6.4



Figure 6.8: cloud report for Resource provisioning scheme with feedback

The policy effectively utilizes VM resources based on job requirements and overall cost is reduced as compared to On Demand policy. However average wait time is somehow increased as Spot VMs were also used to gain economic benefit. Overall cost trend for 10-140 VMs is depicted in the following figure.

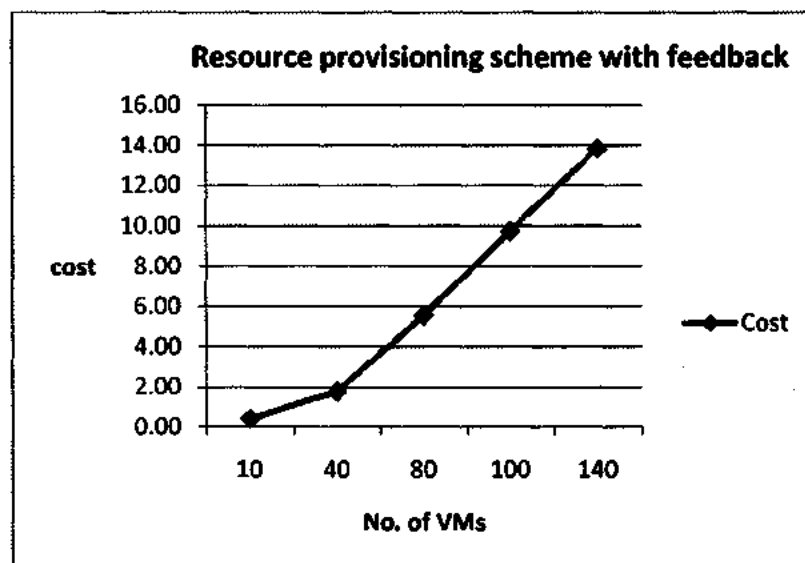
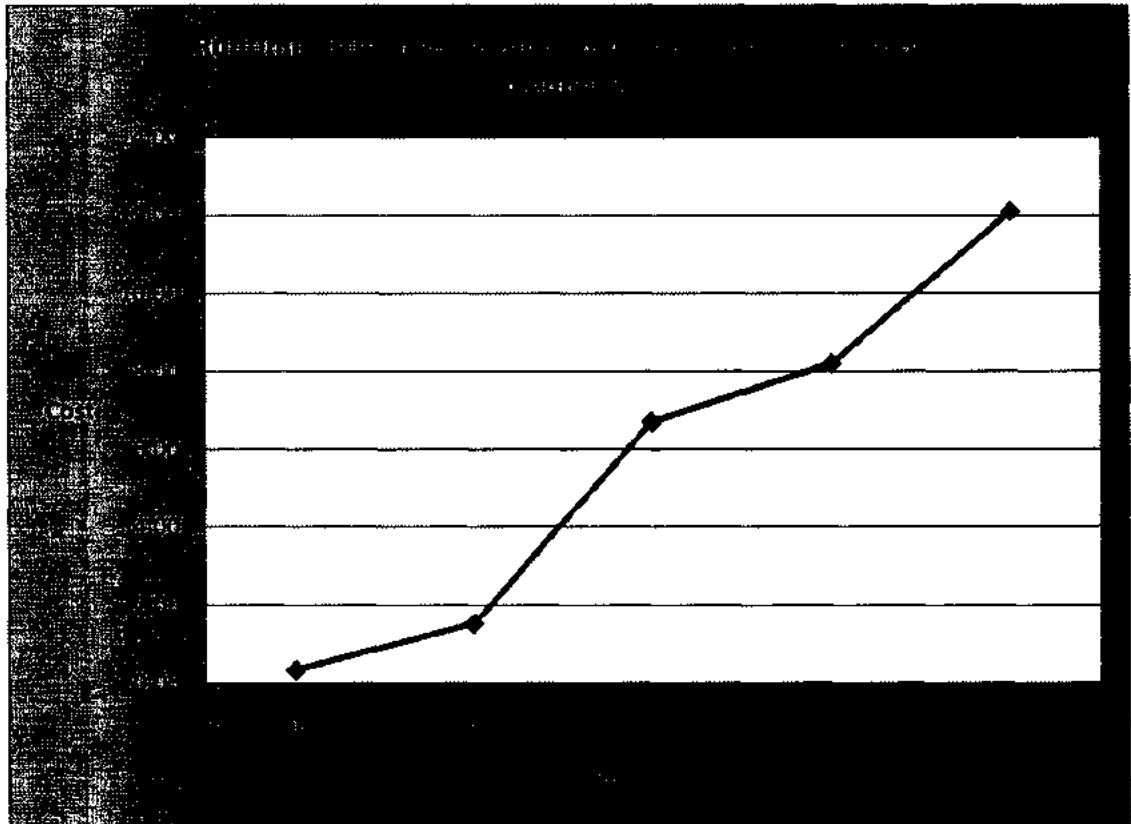


Figure 6. : Plot of Cost-No of VMs for Resource provisioning scheme with feedback

6.4.4 Runtime Estimated Resource Provision Scheme (without feedback)

In this policy we created three threshold limits for job runtime estimation i.e., low, medium and high. Low threshold value indicates that the submitted job is not time bound and the system should reduce cost as much as possible by utilizing Spot instances. High threshold values indicate that submitted job is deadline constrained and the system must use an aggressive strategy to provision the resources with minimal delay. Overall strategy of resource allocation is in accordance with policy Resource Provisioning Scheme with feedback with the exception of job runtime estimation without user feedback. Figure below represents resource utilization of medium threshold limit:



This strategy further reduces overall cost as compared to Resource Provisioning Scheme with feedback. Based on previous literature, it is obvious that user supplied job runtimes are mostly over estimated [17] and hence job estimation runtime can be further optimized as adopted in this strategy.

10	0	0.08	0.24	0.33
40	4	0.31	1.21	1.52
90	28	0.57	6.12	6.69
100	34	0.63	7.57	8.20
140	49	0.80	11.28	12.09

Table.

Overall results of the four policies are presented as follows:

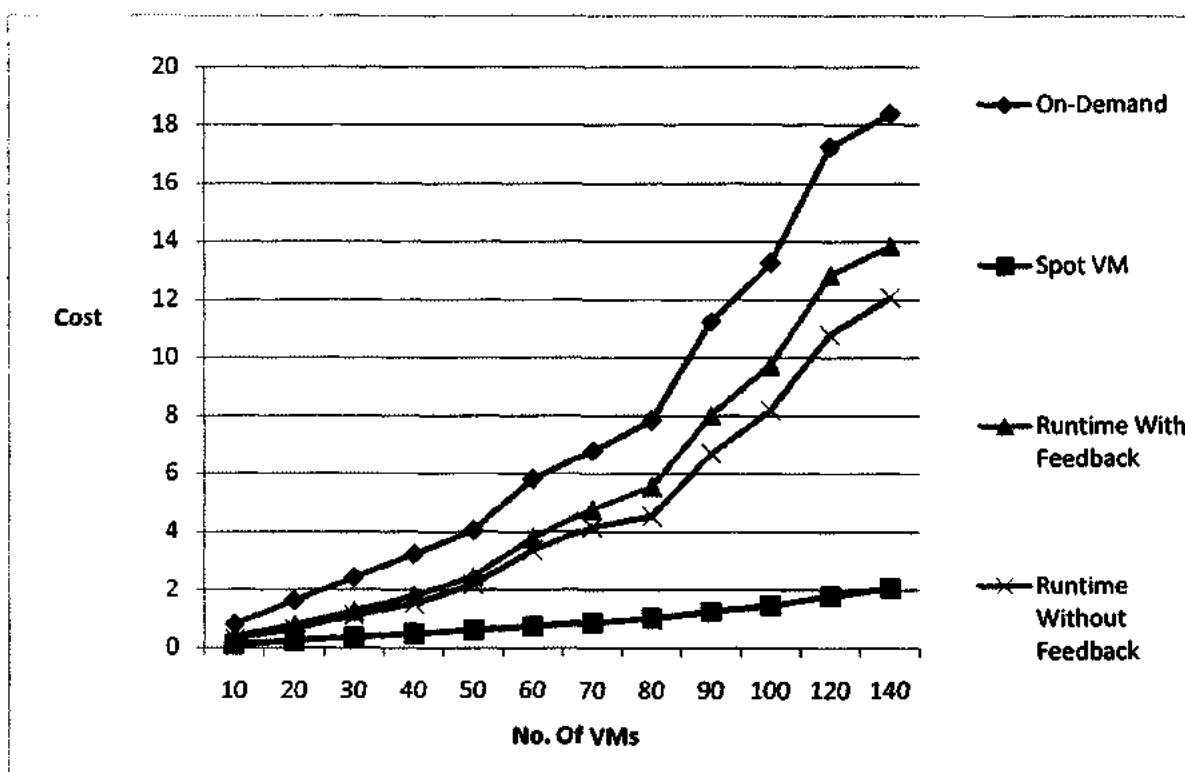


Figure 6. : Overall cost trend for all four policies

As shown in the figure, on demand resource provisioning policy resulted in highest cost to resource allocation. While Spot VM based allocation policy minimized overall cost. Since

these two policies usually result in cost and time overrun respectively. Our proposed policies optimize both time and cost constraints for meeting user specific requirements. Overall Average wait time of four policies are presented in the following figure.

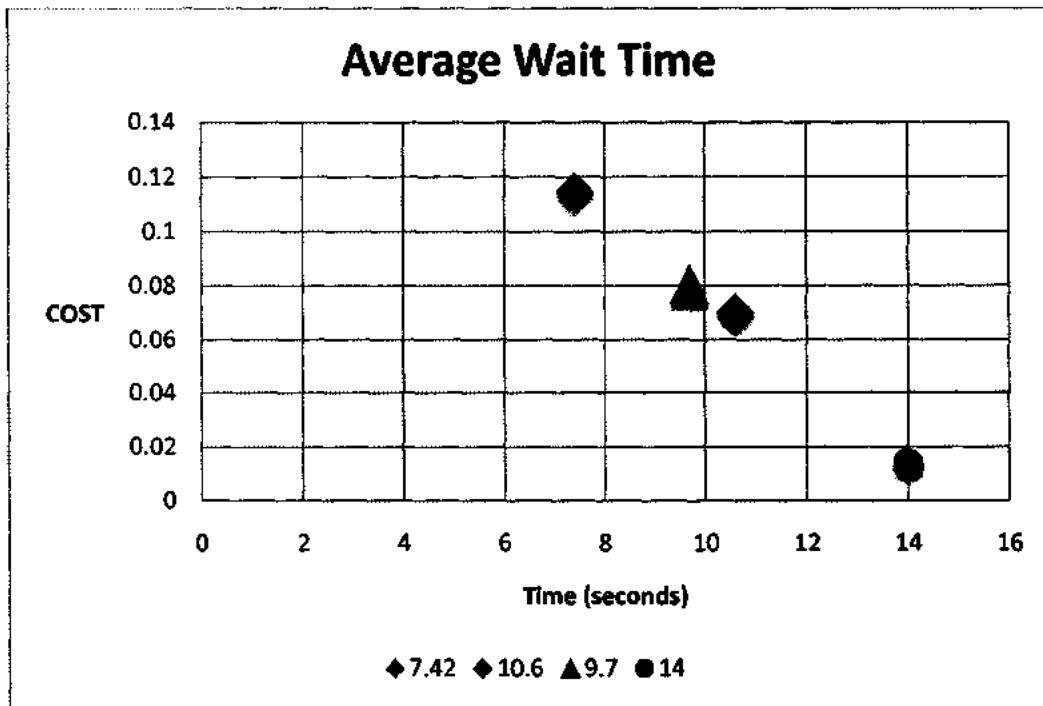


Figure: Average Wait Time of all four policies.

6.5 Evaluation & discussion

CloudSim is a popular cloud simulation toolkit to test resource provisioning policies. Previous studies results indicate that CloudSim simulation results are identical to real world cloud test bed experiments. In this study we introduced four different VM policy and results were compared based on overtime Cost, Time and reliability.

Our results indicate that on demand VM provisioning is not an optimal solution for resource provisioning as it may result in cost overrun. On the other hand Spot VMs, based on its unreliable nature may not be applicable for real time and deadline constrained jobs. Our two proposed policies taking advantages of the above, while minimizing the limitations result in better resource provisioning and VM allocation.

The results also indicate that policy four further optimizes overall resource cost by 15%-20% while hedging against the job failure. Our proposed model can cut down significant amount of VM utilization cost, especially for the jobs of non critical nature.

6.6 Conclusion & Future Work:

The results show promising difference in cost of different policies. Since the simulations were performed to get the results, Real experiments could be performed on the real data acquired by the providers (Google, Amazon and Rackspace). Future workload prediction model strategies could be designed to predict the future workload of the consumer

Options can be further optimized for better and accurate pricing of the resources that helps in deciding to further call or put options.

The proposed algorithm can be improved for more complex or critical jobs. Number of failures can be reduced.

References

- [1] <https://www.gartner.com/doc/2642020/forecast-public-cloud-services-worldwide>
(Hit date=23-03-2014)
- [2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia "Above the Clouds: A Berkeley View of Cloud Computing"
- [3] Buyya, Rajkumar, et al. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." *Future Generation computer systems* 25.6 (2009): 599-616.
- [4] M laden A. Vouk "Cloud Computing – Issues, Research and Implementations" *Journal of Computing and Information Technology - CIT* 16, 2008, 4, 235–246.
- [5] Patel, Pankesh, Ajith H. Ranabahu, and Amit P. Sheth. "Service level agreement in cloud computing." (2009).
- [6] Dillon, Tharam, Chen Wu, and Elizabeth Chang. "Cloud computing: issues and challenges." *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on. Ieee*, 2010.
- [7] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing
- [8] Buyya R, Calheiros R.N, Ranjan R & De Rose C. "CloudSim: A Novel Framework for Modeling and Simulation Cloud Computing Infrastructures and Services." 2010
- [9] Peijian Wang, Yong Qi, Dou Hui, Xue Liu, L Rao, "Present or Future: Optimal Pricing for Spot Instances", 2013 IEEE 33rd International Conference on Distributed Computing Systems.
- [10] <http://aws.amazon.com/ec2/spot-instances/>. (Hit date=23-03-2014)
- [11] Ruiz-Agundez, Igor, Yoseba K. Peña, and Pablo G. Bringas. "A flexible accounting model for cloud computing." *SRII Global Conference (SRII), 2011 Annual. IEEE*, 2011.

- [12] Q. Zhang, E. Gürses, R. Boutaba, and J. Xiao, "Dynamic resource allocation for spot markets in clouds," in Proceedings of the 11th USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE '11), pp. 1–6, 2011.
- [13] Al-Roomi, May; Al-Ebrahim, Shaikha; Buqrais, Sabika and Ahmad, Imtiaz. "Cloud Computing Pricing Models: A Survey", International Journal of Grid & Distributed Computing, 2013.
- [14] D.Allenator, R.K. Thulasiram. G-FRoM:Grid Resources Pricing-A Fuzzy Real Option Model. *3rd IEEE International Conference on s-Science and Grid Computing 2007*.
- [15] D.Allenator, R.K. Thulasiram. Grid Resources Pricing-A Novel Financial Option based Quality of Service-Profit Quasi-Static Equilibrium Model. *9thInternational Grid Computing Conference IEEE, 2008*.
- [16] D.Allenator, R.K. Thulasiram, P.Thulasiram. A Financial Option based Grid Resources Pricing Model: Towards an Equilibrium Between Service Quality for User and Profitability for Service Provider. 2009
- [17] D.Allenator, R.K. Thulasiram. Integrating a Financial Option based Model with GridSimPricing Resources.*10th International Grid Computing Conference IEEE, 2009*.
- [18] D.Allenator, R.K. Thulasiram. Evaluation of a Financial Option based Pricing Model for Grid Resources Management: Simulation vs. Real Data. *12th IEEE International Conference on High Performance Computing and Communication, 2010*.
- [19] D.Allenator, R.K. Thulasiram, P.Thulasiram. Novel Application of Option Pricing to Distributed Resources Management.IEEE 2009
- [20] A. Toosi, R.K. Thulasiram, and R.K. Buyya. Financial Option Market for Federated Cloud Environments. *5th International Conference on Utility and Cloud Computing IEEE, 2012*.
- [21] S. Qanbari, F Li, S Dustdar, and T.S Dai. Cloud Asset Pricing Tree(CAPT): Elastic Economic Model for Cloud Service Providers.
- [22] B. Sharma , R.K. Thulasiram, P.Thulasiram, S. K. Garg. Pricing Cloud Compute Commodities: A Novel Financial Economic Model. *12th IEEE/ACM International Symposium On Cluster,Cloud and Grid Computing 2012*.
- [23] M.R.Rahman,Y.Lu, I.Gupta. "Risk Aware Resource Allocation for Clouds." 2011
- [24] S. Qanbari, F Li, S Dustdar, and T.S Dai. Cloud Asset Pricing Tree(CAPT): Elastic Economic Model for Cloud Service Providers.
- [25] Sharma, B., et al. "Clabacus: A Risk-Adjusted Cloud Resources Pricing Model using Financial Option Theory." IEEE 2014

- [26] J. Hull, *Options, Futures and Other Derivates*. Prentice Hall, 2008
- [27] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, January 1973.
- [28] Ang Li, Xiaowei Yang, Srikanth Kandula, Ming Zhang, CloudCmp: Comparing Public Cloud Providers, IMC 2010
- [29] Dave Durkee, ENKI, The competition among cloud providers may drive prices downward, but at what cost? *Distributed Computing Volume 8, issue 4*, 2010
- [30] Moore's Law vs. Wright's Law, Jim Handy [Online] <http://www.forbes.com/sites/jimhandy/2013/03/25/moores-law-vs-wrights-law>, accessed on August 1, 2014.
- [31] W. Voorsluys and R. Buyya, "Reliable provisioning of spot instances for compute-intensive applications," in *Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications (AINA '12)*, pp. 542–549, March 2012
- [32] W. Voorsluys, S. Garg, and R. Buyya, "Provisioning spot market cloud resources to create cost-effective virtual clusters," *Algorithms and Architectures for Parallel Processing*, pp. 395–408, 2011.
- [33] Q. Zhang, E. Gürses, R. Boutaba, and J. Xiao, "Dynamic resource allocation for spot markets in clouds," in *Proceedings of the 11th USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE '11)*, pp. 1–6, 2011.
- [34] Sudha Srinivasan. "Robust scheduling of moldable parallel jobs", *International Journal of High Performance Computing and Networking*, 2004
- [35] Gerald Sabin. "Moldable Parallel Job Scheduling Using Job Efficiency: An Iterative Approach", *Lecture Notes in Computer Science*, 2007
- [36] He Huang, Liqiang Wang, Byung Chul Tak, Long Wang, Chunqiang Tang, CAP3: A Cloud Auto-Provisioning Framework for Parallel Processing Using On-demand and Spot Instances
- [37] <https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#SpotInstances:sort=requestId>
- [38] Wu, L., Garg, S. K., & Buyya, R. (2012). SLA-based admission control for a Software-as-a-Service provider in Cloud computing environments. *Journal of Computer and System Sciences*, 78(5), 1280-1299

- [39] CloudSim, <http://www.cloudbus.org/cloudsim/>
- [40] CloudReports, <https://github.com/thiagotts/CloudReports>
- [41] Sahar Arshad "Broker based QoS Centric Federated Cloud Framework with Financial Options" (2015)
- [42] Zaman, Sharrukh, and Daniel Grosu. "Combinatorial Auction-Based Dynamic VM Provisioning and Allocation in Clouds." Cloud Computing Technology and Science (CloudCom), Third International Conference on.IEEE, 2011.
- [43] S Ullah "Performance Benchmarking for Infrastructure-as-a-Service Cloud Providers". Manuscript submitted for publication. (2015).

Appendix:

Screen Shots of Cloud Reports:

The screenshot shows the 'CloudReports' application window with a menu bar (File, Settings, About) and a sidebar for 'Simulation environments'. The main area is divided into tabs: General, Hosts, Costs, SAN, and Network. The 'General' tab is active, displaying 'Settings' and 'Information' sections.

Simulation environments:

- New Environment
- CloudReports
 - Provider
 - Rackspace1GB
 - Rackspace4GB
 - Rackspace8GB
 - Google1GB
 - Google7GB**
 - Google13GB
 - Amazon1GB
 - Amazon4GB
 - Amazon15GB
 - Customers
- Run Simulation

Settings

Allocation Policy:	Single threshold		
Architecture:	x86	Upper threshold:	0.8
Operating System:	Linux	Lower threshold:	0.2
Hypervisor:	Xen	VM Migration:	Enabled
Scheduling Interval:	30	Maximal number:	180

Information

Number of hosts:	30
Number of processing units:	480
Processing capacity (MPS):	1200000.0
Storage capacity:	4000000
Total amount of RAM:	195000

The screenshot shows the 'CloudReports' application window with the 'Costs' tab selected. The 'Utilization Costs' section displays four cost parameters with their respective values and units.

Utilization Costs

Processing cost (per sec):	0.1	(Cost per MPS: 1.0410000000000000E-5)
Memory cost (per MB):	0.05	
Storage cost (per MB):	0.001	
Bandwidth cost (per MB):	0.1	

