Clabacus: A Financial Economic Model for Pricing Cloud Compute Commodities

by

Bhanu P Sharma

A thesis submitted to
The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements
of the degree of

Doctor of Philosophy

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
September 2016

Thesis advisor | Author

Dr. Ruppa Thulasiram

Dr. Parimala Thulasiraman | **Bhanu P Sharma**

Clabacus: A Financial Economic Model for Pricing Cloud Compute Commodities

Abstract

Cloud computing at a high level comprises of the availability of hardware, software and technical support via a network protocol to a remote client on a pay-per-use basis. Businesses using Cloud resources has been increasing steadily in the very recent past and the number of Cloud service providers (CSP) are increasing as well. The challenges that characterize a Cloud data center include: on-demand service, elasticity, resources pooling, broad network access, service meters. As the customer base is increasing and their resource requirement and usage pattern has been becoming highly volatile, proper utilization of the resources and generating revenue by appropriately charging the clients for their uses has become an even more challenging research problem. In other words, Cloud resource pricing has emerged as an important and pressing problem to study for ever increasing utility of Cloud computing.

Literature review reveals that there are economy-based models (cash flow, net present value etc.) used for charging mechanism suggested by many researchers. Most of these models are rigid that they are not build with the core of Cloud - elasticity - in mind. Also, the economic models do not provide flexibility of the economy of scale to either increase or decrease the resource requirement and appropriately charge for such increase or decrease in resource use.

For my thesis, I have designed and developed a Cloud resources pricing model that satisfies two important constraints: the dynamic ability of the model to provide a high satisfaction guarantee measured as Quality of Service (QoS) - from users perspectives, and profitability constraints - from the Cloud service providers perspectives. I have employed financial option theory and treated the Cloud resources as underlying assets to capture the realistic value of the Cloud Compute Commodities (C3). I have priced the Cloud resources using my model.

Through this research, I show that the Cloud parameters can be mapped to financial economic model and that this model can be effectively implemented for resource pricing purpose. I discuss the results of pricing Cloud Compute Commodities ($C3$) for various input parameters, such as the age of the resource and quality of service.

Contents

List of Figures

List of Tables

Acknowledgments

Firstly, I want to thank my advisors, Dr. Ruppa Thulasiram and Dr. Parimala Thulasiraman, for their able guidance during my research and study at University of Manitoba. My very sincere thanks to them for their support, encouragement and great effort they put into training me during the course of this thesis.

I would like to thank my father "Mr Ashok Sharma", my wife "Dr. Mansi Aparnathi Sharma" and my brother "Mr Keshav Sharma" for their emotional and moral support. Everything that I have achieved so far can be attributed to their love and sense of security they provided to me.

Finally I would like to thank all the graduate students at the Department of Computer Science for making it a very friendly place to work. I want to thank them for all their help and support.

This thesis is dedicated to loving memory of my late mother , "Dr. Aruna Sharma". I remember her for all her kindness and wisdom

Chapter 1

Introduction

The era of modern computing can be traced back to 1837 when Charles Babbage (Hyman [1985]) first designed a computing device. This device had essential memory, arithmetic and logical processing units. Significant advances in electronics were required to design the first computer, which was not attained until 1930s (Ceruzzi [2003]). Over the next few decades, with the advances in transistors and microchip processors saw the age of desktop computers. By the 1980s, the use of computers was widespread in industry and academia. With the Internet revolution of the 1990s coupled with the success of Microsoft Windows, computers became household devices and attracted millions of users across the globe. Computing needs increased many-fold over the years, which led to the advent of distributed computing and eventually to the Grid computing in early 2000s.

A Grid is a collection of geographically distributed resources (Foster and Kesselman [1999]) that enables high performance computing in a dependable, inexpensive, and consistent manner. In simpler terms, a Grid is a collection of hardware and software services, which a consumer utilizes like any other utilities such as the electrical power supply, water etc. A user can connect to a Grid using Internet and use the services without being aware where the physical resources are located. This is the main idea behind Grid computing. In general, a Grid can support multiple operating systems, varying memory requirements and customer's specific security requirements. There are five major focus areas in Grid (Foster and Kesselman [2003]) :

1. Resource sharing: The Grid resources have been used effectively in scientific studies involving scientists from different geographical locations. This is made possible by simplifying the use of resources by different people. The access to remote hardware, software and database is the prime focus area of grids.

2. Security: The Grid resources are exposed to a wide range of people and institutions. In such a scenario the security and privacy of the data and software is of paramount importance. Access policy, authentication and authorization form the core of security in Grid resources.

3. Resource usage: At any given time multiple users are running their jobs on the Grid. The efficient utilization of the Grid resources is very important to accommodate more users and jobs. The backend scheduler does the task of efficiently scheduling the jobs and the user is completely oblivious of this process.

4. Easy access or locality: With the advances in networking technology the users can submit their jobs remotely to a Grid. Locality of Grid resources makes it possible for a user without knowing the geographical location of the Grid resource, to interact with the Grid seamlessly as a local machine.

5. Open standards: Grid resources are meant to be used by people following different standards, e.g. programming standards and naming conventions. It is the responsibility of the Grid administrator to accommodate all the users and also provide a platform for uniform and easy access.

A Grid was used mostly to provide a platform to the end users; i.e. users could request a particular architecture to run their jobs. Eventually more specific software demands were made by the users, which eventually gave rise to a concept of Software as a Service (SaaS). These softwares were owned, delivered and managed by the software vendors and the users paid for these services on a pay-per-use basis. This initiated the idea of utility computing which eventually led to the advent of Cloud computing. In simple terms, the combination of Grid, SaaS and utility computing can be defined as *Cloud computing*. The hardware and utility providers propelled the use of Grid computing, while the Cloud software vendors opened up the market for Cloud computing.

Grid computing focuses on collaborative and distributed shared resources while Cloud computing concentrates on providing services for users on a pay-per-use basis. One other major development in Cloud computing is the virtualization of resources. Improvements in virtualization and increasing demand for SaaS from business communities led to the exponential demand for Cloud resources. Grid computing traditionally attracted scientist groups for projects like Large Hadron Collider (LHC) at CERN while Cloud computing engrossed industrial projects like Salesforce which provide Customer Relationship Management (CRM) services. CRM is now widely used to manage client information, track sales and find prospective customers. Providing seamless service to clientele has raised many issues that Clouds share with Grids, for example, resource sharing, security, resource usage, data locality etc. There are many research works reported in Cloud computing literature addressing these issues. However, there are few other issues such as virtualization and resource pricing that are unique to Clouds. It is clear from literature that virtualization has received a lot of attention from researchers and there has not been much work on resource pricing.

Due to the commercial interest in Cloud computing, the pricing of Cloud resources remains a challenging problem, both for Cloud vendors and clients. Cloud vendors like Amazon EC2 (Amazon)and Rackspace(Rackspace) provide a price list of various resources, but the underlying pricing policy is not revealed to the end users. For clients to make an informed decision, it becomes necessary to have a better understanding of the pricing policies. A small business might be interested in knowing if it is beneficial to develop an in-house IT infrastructure or to lease the Cloud resources from a Cloud vendor. Better and transparent pricing policies can help clients in evaluating the true cost of Cloud resources. Some attempts have been made to price Cloud resources using economy-based models, but these models cannot incorporate the elasticity of Cloud resources. These are rather simple models which are not capable of capturing all the complexities of Cloud pricing. Finance based models could be used to overcome the limitations of economy-based models, which could provide more flexibility in pricing Cloud resources.

Contributions of this research can be divided into four parts. (i) Design and development of algorithms for financial models for option pricing such as Black-Scholes-Merton model and Binomial lattice etc. and their use in Cloud resource pricing, (ii)

Cataloguing and characterizing the cost parameters for Cloud computing (Cloud input parameters); this is done to determine the cost incurred by the vendor to set up a Cloud (data center) and to study their impact on the retail price to clients, (iii) Mapping these input parameters to the financial models for option pricing and (iv) finally developing a software architecture framework (called Clabacus) to evaluate the Cloud Compute Commodities ($C3$). This architecture could be used by vendors and clients to compute optimum price of Cloud resources for any specific tasks.

The rest of the thesis is presented as follows: in Chapter 2, I discuss the Cloud literature focusing on resource pricing; Chapter 3 covers my solution methodology. I discuss option pricing techniques in Chapter 4 and the Clabacus architecture is discussed in detail in Chapter 5. I present Value-at-Risk analysis in Chapter 6. I discuss experiments and results in Chapter 7 followed by conclusions and future work in Chapter 8.

Chapter 2

Literature review

In this Chapter I will explain the basics of Cloud computing. I will start with some fundamental definitions of Cloud computing.

According to RAD lab at UC Berkeley Armbrust et al. [2009]; "Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). The datacenter hardware and software is what we will call a Cloud."

Gartner Inc. (Plummer et al. [2008]) defined a Cloud as "a style of computing where massively scalable IT-enabled capabilities are delivered 'as a service' to external customers using Internet technologies."

Foster et al. (Foster et al. [2008]) defined Cloud as "A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet."

The Cloud lab at University of Melbourne (Buyya et al. [2009]) defined Cloud as "A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on service-level agreements established through negotiation between the service provider and consumers."

According to the National Institute of Standards and Technology (Mell and Grance [2011]) "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."

Comparing these definitions, Cloud computing at a high level comprises the availability of hardware, software and technical support via a network protocol to a remote client. Also, as I mentioned before, Grid computing along with utility computing and SaaS gave rise to Cloud computing. The focus area of Cloud are closely related to that of the Grid and are discussed next.

2.1 Focus areas of Cloud computing

As explained in the previous section, Grids were/are mostly used by the scientific community while Cloud computing was driven by the software industries rendering their services to clients and businesses that do not want to spend on infrastructures and maintain these infrastructures. Grids were conventionally used for jobs with large computational needs, while the Clouds are mostly used for large numbers of simple jobs. Clouds have not yet been extensively used for high performance computing (HPC) and the future of scientific computing on Clouds would depend on the Cloud vendor's ability to incorporate compute intensive jobs. Secondly, Grids were developed by organizations like governments, national labs etc., for academic and research purposes and motive has not been for commercial purposes or for earning profit. Therefore, financial aspects like billing, though considered, were not primary focus. Grid focus has been on sharing resources efficiently. On the other hand the Clouds have been set up for commercial purposes and hence pricing the Cloud resources efficiently is a very important area of research. While research on use of Clouds for HPC is still in it's infancy, in Woitaszek and Tufo [2010] the authors discuss the billing of HPC using Cloud resources. Some key focus areas of Cloud computing as put forward by NIST (Mell and Grance [2011]) are:

1. On-demand self-service: The client should be able to connect to the remote service provider without requiring any assistance. Through virtualization, the client should be completely oblivious to the geographical location of the requested resource.

2. Broad network access: The clients can access the remote services through the cutting edge networking technologies and these services should cater to the multitude of client requests. In other words the Cloud resource vendor should be able to provide the services irrespective of the platform or operating system used or demanded by clients.

3. Resource pooling: To maximize the use of resources, the Cloud resource vendors use multi-tenant model in which the real and virtual resources are assigned and re-assigned to a large number of clients. Various scheduling and resource allocation algorithms are used by the vendor to make it efficient and the client should be unaware of this process. Amir et al. (Amir et al. [1998]) present a cost-benefit framework for efficient resource pooling.

4. Rapid elasticity: One of the key difference between the Grid and Cloud computing is the Cloud's elasticity to incorporate the increased demands of the clients. The clients do not have to worry about requiring additional computational resources in real time, as the elasticity of the Cloud will be able to take care of it to a great extend.

5. Measured service: The Cloud system efficiently monitors the resources used by the clients. This monitoring provides transparency to service providers and clients. The client is oblivious to the pricing models used by the vendor and the vendor might overcharge the client. To avoid this, Carlos et al. (Carlos et al. [2011]) developed a bilateral pricing model in which both the client and the vendor monitor the usage independently.

Besides these focus areas, Cloud can be characterized by service and deployment models (for example Mell and Grance [2011]).

2.2 Service models

1. SaaS : Services like Gmail or Salesforce (Salesforce) fall under this category. The software is managed and monitored by the vendor, and the service is rendered to the client via web-based tools or some other program interface.

2. PaaS : Services like Mosso (Mosso) that provide clients with servers or storage space fall under this category. In PaaS the client has more control over the Cloud resource than in SaaS. The client can change the application hosting environment by changing some application settings which is not available in SaaS.

3. IaaS : This gives the clients even more control over the Cloud resources. The clients can change the underlying operating system or install the needed softwares. Amazon EC2 (Amazon) is an example of IaaS.

Cloud can also be classified as public, private or hybrid. A private Cloud can be accessed only by clients belonging to a single organization or business. For example, all employees of a business share resources in a Cloud model. This is mainly done to keep some proprietary computations in-house and avoid exposure to competing businesses. A public Cloud is a data center that can provide service to any number of clients or businesses. The hybrid Cloud is a collection of private and/or public Cloud which might have the common technology or scheduling policies, which helps in load balancing between the Clouds (Mell and Grance [2011]). For example, sensitive computation of a business can be done in-house with a private Cloud while general computations can be done in a public Cloud. These classifications aside, the philosophical meaning of a Cloud - virtualized service to clients over the Internet - is generally violated when resources are made available to clients belonging to a single organization only. Hence, classification of a set of resources as “private Cloud” is a subject of serious debate in the literature.

One of the major Cloud resource vendor Amazon(Amazon) offers various services and the clients are free to choose between on-demand, reserved and spot instances of any architecture they want to use. Clients who need some computational cycles urgently for a short duration could buy on-demand instances by the hour. In a reserved instance a client pays a one-time premium to reserve a resource and receive a discount on the services when they use it, therefore reserved instances are cheaper than on-demand instances. To utilize the unused computational cycles or storage space, Amazon(Amazon) issues spot instances. These instances are issued on an hourly basis and bidding is done by clients to acquire these instances. The spot prices change every hour based on the availability of resources and if a client wants to continue to run their jobs, they have to bid higher than the current spot price. The detailed listing of various Amazon instances can be found on Amazon pricing. Anandasivam and Premm (Anandasivam and Premm [2009]) present a bidding and a dynamic resource pricing models to maximize the profits of the Cloud vendor.

Although Cloud resource vendors provide the detailed price list of various instances, the clients do not know the underlying pricing policies. Clients do not know if leasing a resource from a vendor is more cost efficient than buying that resource. So how to ensure that the prices offered to clients are fair? How vendors can make sure if they are earning the desired profits? To answer these questions, pricing of Cloud resources is very important. Through this research, I develop a transparent pricing model that can answer the above-mentioned questions. In the rest of this chapter, I describe economy-based models that could pave a way to price Cloud resources.

2.3 Economy-based pricing models

Net present value (NPV) and discounted cash flow models (Shrieves and Wachowicz-Jr. [2001]) are two important economics based pricing models. In this section I will explain them briefly.

2.3.1 Net present value

A cash outflow is the investment made to acquire some asset; like a Cloud vendor would invest some money in setting up a new service. This money spent in setting up a new service comprises of a series of cash outflows. A cash inflow is the money that the clients would be paying to the vendors for using their services. The difference between the inflow and outflow value is known as the net present value or net present worth (as explained in Investopedia). In some cases the only cash outflow is the initial investment and all the future cash flows are inbound; in such cases the NPV is calculated by calculating the present value (or the discounted value) of the future inflows and subtracting the initial investment or cash outflow from it. In cases, where there are series of inbound and outbound cash flows in future, all cash flows are discounted to the current date to evaluate NPV; the present value of all outflows is subtracted from the present value of all inflows.

The discounted cash flow model uses the NPV model and is explained in the next section.

2.3.2 Discounted cash flow model

The discounted cash flow model is used to track the movement of cash in and out of a business. All future cash flows; inbound or outbound are discounted to get their present value. The difference between the present values for inbound and outbound cashflows represents the cash holding of a business.

2.3.3 Cloud resource pricing - problem statement

Net present value and Discounted cash flow models are very simple economy-based models and can be very easily used to price Grid and Cloud resources but they cannot incorporate the elasticity of Cloud resources. In Walker [2009], the author analyzes the true cost of leasing a CPU for an hour against acquiring and owning the same CPU. This study concludes that financial option based pricing would be an appropriate technique for Cloud resource pricing. For my research, I develop a financial economy based pricing model.

Authors in (Patel and Shah [2005]) explore the cost incurred by data centres. This study focuses on three major issues: space, power and cooling for the cost model. They provide a step by step analysis of the cost for each of the three issues and add these costs to obtain a comprehensive cost of running a data center. The authors of this study do not go any further in finding the cost of Cloud resources meant to be sold as a service.

The collaboration between different Cloud vendors to provide diverse resources to clients is known as a federated Cloud. In Mihailescu and Teo [2010] the authors introduce a dynamic strategy-proof pricing scheme to incorporate diverse resource requests from the Cloud vendor to the federation. Teng and Magoules [2010] build a dynamic billing and allocation policies. Through these policies the user can predict the future Cloud resource prices and adjust their budget accordingly. Macas and Guitart [2011] used genetic algorithms to price Cloud resources based on the underlying rule that the prices of the Cloud resources may fluctuate based on their usage. This study, however, did not explore the Cloud input parameters in details and was mostly aimed at developing a Genetic algorithm for pricing. A recent study on the economy of spot instances by Wee [2011] suggests that the additional cost savings to the businesses for moving the workload from off-peak periods to spot instances is not large. This cost savings is achievable under certain ideal conditions, for example, on scheduling for dispatching tasks. Their study looked at one year of Amazon's spot instance price data to arrive at this conclusion.

In order to understand the price fluctuation of Amazon spot instances, Javadi et al. [2011] explored to fit a statistical model to the existing prices. Using the proposed model, the authors have done a comprehensive analysis of spot instances based on one year price history in four data centers of Amazon's EC2. The authors showed that the statistical model they have proposed fits well with these data series and claim that they would be able to model the dynamics of spot price.

In another work, Toosi et al. [2011] explored ways of increasing the profit for Infrastructure as a Service (IaaS) providers by increasing the resource utilization. The authors studied this problem for a Cloud provider within a “Cloud federation” and they suggested several policies to increase utilization based on the resource prices at other providers within the federation. Fundamentally, this study focuses on increasing the profit for a provider through higher utilization of resources.

All these studies have focused on investigating the existing prices or how to derive cost savings for the users based on current prices mostly for spot instances of Amazon EC2. To the best of my knowledge, devising a quantitative approach to compute the price of Cloud resources has not been the subject of serious investigation so far.

Problem Statement: Designing and developing an architecture/model to price Cloud compute commodities that would strategically capture the financial aspects of operating a data-centre and the technological developments that encompass the hardware/software systems that comprise a data-centre.

There are few studies in the recent past that explored the use of the financial option concept for pricing technology resources, for example Allenotor and Thulasiram [2008] for pricing Grid resources; Kumar et al. [2011] and Singh et al. [2013] for pricing transmission rights in power systems. These studies consider financial option theory in its original form, which cannot be directly extended to pricing Cloud resources. This is because of the price fluctuation in Cloud instances as well as fluctuation in their availability.

Rahman et al. [2011] used the American option to evaluate the spot price of a

Cloud resource. The authors did not discuss the Cloud input parameters. They also did not consider some important factors like the rate of depreciation of the resources and the quality of service to the clients.

For my research, I have used the financial option pricing models to price the Cloud resources and incorporated various input parameters. I have considered factors like the rate of depreciation of the resources and the quality of service and significant part of my thesis has been published. For example, in Sharma et al. [2012b], I employed other known concepts such as Moore's law (for the quality of the resources in Cloud) and combined it with an interest rate formula to price Cloud resources for current market conditions. Hence, this work is unique in many fronts for the Cloud services. The focus of this study was to price Cloud resources from the vendor's perspective. In Sharma et al. [2015b], I generalized my pricing algorithm to incorporate both Client and Vendor side pricing. Another unique strategy of my research is the use of Value-at-Risk to refine resource pricing.

In the next Chapter, I present and explain my solution methodology for the problem in detail, that includes both financial and technical aspects of a Cloud data-centre.

Chapter 3

Solution Methodology

In this Chapter I will explain my thesis in the form of my strategies to do research that would help in pricing Cloud Compute Commodities ($C3$). To achieve this task, I will first introduce some essential features from both computer science and finance. In particular, I will explain some basic finance concepts first. Then, I will present Moore's law, a technological aspect of computing that predicts the trend in hardware technology. How I strategically combine these concepts from two totally different fields is one of the core contributions of my thesis. Designing and developing an architecture using this contribution is done for rest of my thesis work as explained in chapters 5 through 7.

3.1 Finance Market

One of the essential activities in a finance market is trading of equities, commonly known as stocks. There are thousands of stocks representing various business sectors that are traded in various exchanges around the globe. These trades determine the economic well being of a nation, based on which Governments decide on various policies such as trend setting interest rates.

Trading of stocks in financial market is explained using Figure 3.1.

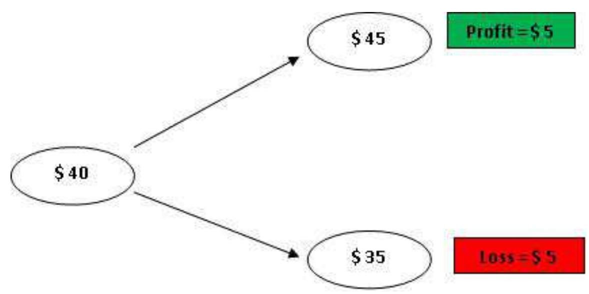

Figure 3.1: Stock trading

Assume a person buys a stock at $40 at a given time. Let us assume the price goes up to $45 in a near future time. At this point, the owner of the stock may sell it to make a profit of $5. On the other hand, if the price of the stock falls to $35 during that period the owner would have to incur a loss of $5 on the same stock if the owner decides to sell it. Therefore, by owning assets (that is, buying stocks in this example) always come with a risk of losing value of the assets depending on the market downtrend. A way out of this conundrum has been the subject of research for several decades in the 20th century and in 1973 a risk-neutral strategy of investments was put forth that opened up a new multi-billion market on derivatives market (see for example, Thomsett [2009]).

Derivatives are financial instruments that are derived from some fundamental assets. For example, options are derivatives with stock and other financial instruments as underlying assets and futures are derivatives that have agricultural commodities as underlying assets. While the concept of derivatives has been around for a few centuries, the use of options to eliminate risk was put forward by Bachelier [1900] in his PhD thesis. Financial derivatives (such as financial options) creates an opportunity to opt out from the contract to avoid losses while allowing to lock-in the profit making opportunity. That is, the risk involved in investing is eliminated to a large extent with simple financial options. However, there is higher risk of losing lot of money through the premium paid while entering the option contracts when options expire without getting exercised since the volume of trade in such contracts is usually very high. Details of option trading is presented in the next section. Appropriate use of financial options is an early focus of my thesis study for pricing Cloud compute commodities.

3.2 Option trading

Option trading became organized and prominent in the early 1970s and with the establishment of Chicago Board of Options Exchange in 1973 [1], it started attracting many investors. Prior to this, few options were traded by the insiders in the exchanges, but there were no specific rules or valuations. A financial option is a contract which gives its owner (holder) a right to buy or sell the underlying asset. It is a right and not an obligation. The other party in the contract (writer) is obligated to the decision of the holder (Hull [2011]).

There are two types of options: a call and a put option. This classification was first introduced by Russel Sage in 1872 (Thomsett [2009]) for US markets. A call

[1] CBOE - Chicago Board of Options Exchange: "www.cboe.com/aboutcboe/history.aspx"

option gives its holder a right to buy an underlying asset (e.g. stock) as per the contract. Let me introduce an example to explain option contracts. Let us assume that the stock price of an underlying asset at the time of contract is $40 and the strike price - the price at which the holder may exercise (that is, buy (with call option) or sell (with put option)) the option is $42. If the stock price increases to $45 during the contract period, the holder may exercise the option (that is buy the stock at $42 with a call option) and sell the stock in the open market at $45 and make $3 for each stock. This scenario can be explained as shown in Figure 3.2. As it is a right and not an obligation, if the stock price falls, the holder of that option may decide not to exercise it, thus providing immunity/leverage against any potential loss, other than the premium paid by the holder at the start of the contract.

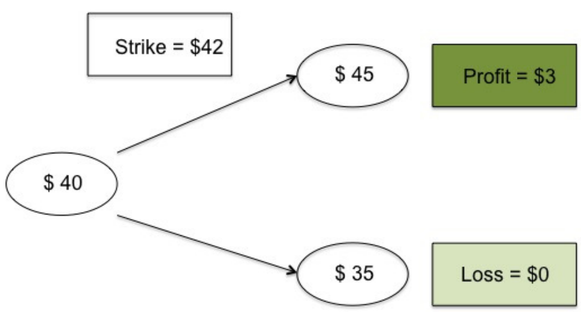

Figure 3.2: Option trading

There are many styles in which these two types of options can be exercised, the simplest being European and American options. An European option can be exercised only at the time of expiration of the contract while an American option can be exercised any time before the expiration. There are many exotic options like Asian, Bermudan, Barrier etc., (Hull [2011]). A typical option would comprise of a type and style; couple of examples could be European Call option or American Put option.

Finding the worth of an option contract is known as the option pricing problem. Essentially, option pricing is to compute if the premium being asked for the option contract is worth for the particular contract being decided upon.

There are five essential parameters/variables in pricing an option. S is the current stock price, K is the strike price i.e. the buy/sell price set for the stock in the contract, T is the expiration time (contract duration), r is rate of interest and σ is the volatility of the underlying asset. Volatility is a measure of fluctuation or uncertainty of the stock price during the contract period. If we know these five parameters, we can evaluate the option price using any of the option pricing techniques explained in Chapter 4.

In generic terms, a financial derivative (such as an option) is a contract between two organizations, two investors or an investor and an organization which specifies some conditions. Some of these conditions are expiration date, execution (strike) price, exercise time (when to buy/sell) etc. As I mentioned earlier, a financial derivative derives its value from some underlying asset, for example, options derive their value from underlying stock prices. Some other types of derivatives are commodity derivatives, which derive their value from commodities like food grains or metals; interest rate derivatives, which derive their value from rate of interest offered by banks and currency derivatives which have currencies as their underlying asset.

Due to its complex mathematical nature, a closed-form solution (that is, formula for finding option value for a given parametric conditions) is available only for some simple options such as European Call option. for all other styles of options, one has to resort to numerical techniques. Pricing an option using numerical techniques is a computationally intensive problem and numerical techniques from varying disciplines have been attempted to price option in the past. In Chapter 4 I present a couple of such techniques used by researchers in finance.

3.3 Moore's law

Gordon E. Moore (Moore [1965]) in 1965 stated that "the number of transistors that can be placed on a circuit will double roughly every eighteen months". This statement has been observed to be true so far and hence attained the status of a *law*. This law seems to holds true for processing power, memory etc. This law can be presented as:

$$ProcessingPower_{t=T} = ProcessingPower_{t=0} \times 2^{T/2} \tag{3.1}$$

We can use Moore's law to estimate the improvements in hardware design, but this law cannot be used directly to estimate the price of new hardware (especially for the purpose of building a Cloud data center) because to price, some other factors such as rate of inflation need to be taken into account.

3.4 Compound interest

The $FutureValue$ of an asset can be evaluated using the $PresentValue$, the rate of interest (r) for the period of years (n) using the well known formula (as explained in Investopedia Investopedia) [2].

[2]Investopedia: www.investopedia.com/walthrough/ corporatefinance/3/time-value-money/future-value.aspx?0=40186&l=dir&qrsc=999&qo= investopediaSiteSearch&ap=investopedia.com

$$FutureValue = PresentValue \times (1 + r)^n \tag{3.2}$$

Bringing this to the context of my thesis, $PresentValue$ can be equated to the initial investment by a Cloud service provider (CSP) in building a Cloud data center and the $FutureValue$ is the initial investment's worth at the end of the contract period. That is to say that the "interest rate" can be mapped to the rate of change (depreciation, in my case since by nature computer hardware tends to lose value quicker than other assets) of the investment on infrastructure.

3.5 Compounded-Moore's law

In essence, Moore's law covers the technical aspect and compound interest covers the financial aspect of the investment in infrastructure. For pricing $C3$, depreciation of the existing infrastructure, inflation, and technological evolution based on Moore's law all have to be considered together. For this purpose, I introduce a new formula derived from two distinctly different concepts, which I call Compounded-Moore's law is presented below.

$$XResourceVal_{t=T} = XResourceVal_{t=0} \times (1 + r)^{T/2} \tag{3.3}$$

This equation computes the depreciation of a Resource X based on Moore's law. However, in conjunction with compound interest formula presented in Equation 3.3, the value of a resource X is computed indirectly through this equation as well. In other words, the single Equation 3.3 captures the technical and financial aspects of the resource.

3.6 Use of option pricing to price Cloud resources

I identify the Cloud parameters and map them to financial option parameters in order to develop resource pricing algorithms (Algorithm 1 and Algorithm 2).

The input Cloud parameters used for Cloud resources pricing are:

1. *Capital Investment* (I_C): This gives the Cloud service provider's expenditure per year. For example, a service provider might buy a resource X each year. According to the Compound-Moore's law, for a given investment duration, the provider will reap more processing power at a constant price. Also, the service provider will pay less amount to buy the same resource X next year and even lesser in subsequent years. When pricing the resources from the Clients' perspective the capital investment is the estimated initial investment that the client would incur to install and own such a resource.

2. *Contract time* (T): The time period the client wants to lease the resources from the Cloud service provider. From the client's perspective, this could relate to the actual *use-time* of the resources for pricing.

3. *Rate of depreciation* (θ): It is the rate at which the infrastructure of service provider is expected to lose its value, both financial and technological. The pricing policies of service provider should be such that they make profits on their initial investments before the clients no longer want to lease these resources. The information about the age of resources generally may not be available to the

clients and hence while pricing from the client's perspective this parameter is an estimate.

4. *Quality of service* (QoS) (r_s): This is the quality assurance from the service provider to the client. This could include the turnaround time, accuracy of results, data privacy and contingency plans etc. QoS is the primary criterion while pricing the resources for services from both the provider and clients perspective.

5. *Age of resources* (T_{res}): It represents the age of a particular resource the service provider is leasing to the client. The start time of a particular task in a resource in conjunction with the age of the resource affect the price for the services.

3.6.1 Mapping Cloud parameters to option pricing parameters

As discussed before, to evaluate/price a financial option we need to know five input parameters S (initial price), K (strike price), r (interest rate), T (expiration time) and σ (volatility). $Total\_investment$ is the total money that the service provider will spend during the lifetime of a contract and its value can be calculated using Compound-Moore's Equation 3.3 with $Initial\_investment$ as one of the input parameter. $Strike\_estimate$ is the equivalent of K, evaluated using Compounded-Moore's formula (Equation 3.3) with $Contract\_time$ and $Rate\_of\_depreciation$ as input parameters. Similarly $Volatility\_estimate$ is the equivalent of σ with $Age\_of\_resources$ and $Rate\_of\_depreciation$ as input parameters. With this mapping, any option pricing technique as explained in Chapter 4 can be used to price a Cloud resource. The algorithms for pricing Cloud Compute Resources are given below.

Algorithm 1 Pricing Cloud resources

Get the input Cloud parameters

$I_{C_{total}} = Compounded - Moore(T, I_{C_{initial}})$.

$K_{est} = Compounded - Moore(T, \theta)$

$\sigma_{est} = Compounded - Moore(T_{res}, \theta)$

Map Cloud parameters to option parameters

$S \Leftarrow I_{C_{total}}$

$K \Leftarrow K_{est}$

$r \Leftarrow r_q$

$t \Leftarrow$T

$\sigma \Leftarrow \sigma_{est}$

Use any option pricing technique as explained in Chapter(4) to price the Cloud resource.

Algorithm 2 Compounded-Moore (T,a)

{

$X = a \times (1 + r_q)^{T/2}$;

return X;

}

Chapter 4

Option Pricing Techniques

In this chapter I will explain some traditional option pricing techniques that I have used in my thesis. These option pricing techniques can be used to price $C3$ using Algorithm 1 as explained in Chapter 3. In one of my works (Sharma et al. [2012b]) I used the classical Black-Scholes-Merton (Black and Scholes [1973], Merton [1973]) model and mapped $C3$ to the underlying asset in this model.

In this Chapter, I present the fundamental concepts from the closed-form solution given by Black-Scholes-Merton model. This model is good for simple contracts in Cloud market. To handle complicated contracts the Black-Scholes-Merton model is not sufficient and hence I have used numerical techniques such as binomial lattice and other heuristics. I have introduced complications (reality) to $C3$ contracts to simulate the needs of small, medium and large enterprises. For example, a large enterprise may have dynamic business strategies and its computing needs would fluctuate a lot. That is, resources necessary for such business would evolve over a period of time. The $C3$ contract should accommodate such evolutions. Beyond simple options, I cannot map

$C3$ contracts directly to finance models since such models do not exist. However, I have revitalized fundamental numerical techniques that I present in this section to study such complicated contracts in the Cloud market. These techniques along with other pricing techniques such as finite-differencing (for example, Tavella and Randall [2000]), Fast Fourier Transform (for example, Carr and Madan [1999]), Ant Colony Optimization (for example, Kumar et al. [2009]), Particle Swarm Optimization (for example, Thulasiram et al. [2014]), form an important module (computational block) in my proposed architecture explained in Chapter 5.

4.1 Black-Scholes-Merton model

Fisher Black and Myron Scholes(Black and Scholes [1973]) and Robert Merton (Merton [1973]) formulated a model for option pricing in 1973, which was presented as partial differential equation. This model revolutionized the option market and received the Nobel prize in Economics in 1997. This mathematical model has been applied on simple European call and European put style options to obtain a closed-form solution. That is, if we know the five input parameters for options, we can find the option value using the closed-form solutions of Black-Scholes-Merton (BSM) model.

For all other styles of options, exact (closed-form) solution could not be found due to complex nature of these options and mathematical models representing these options. However, approximate solutions for various options such as complex chooser, arithmetic mean , time switch options can be obtained from the Black-Scholes-Merton model.

The classical Black-Scholes-Merton closed-form solution for a European call option is given by (see for example, Hull [2011]).

$$C(S,t) = N(d_1) \times S - N(d_2) \times K \times e^{-r(T-t)} \tag{4.1}$$

where,

$$d_1 = \frac{\ln(S/K) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{(T-t)}}, and \tag{4.2}$$

$$d_2 = d_1 - \sigma\sqrt{(T-t)} \tag{4.3}$$

This formula for a put option is

$$P(S,t) = N(-d_2) \times K \times e^{-r(T-t)} - N(-d_1) \times S \tag{4.4}$$

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 and t is the expiration time. $N(d)$ represents the normal distribution function on d.

Although the Black-Scholes-Merton model transformed the art of option pricing, it has some drawbacks. First, the closed-form solutions are applicable only for simple European call and put options. Secondly, this model assumes constant market volatility (σ), which does not reflect the real market scenario. Due to these and other limitations, such as the underlying asset price distribution, research turned to other pricing techniques.

4.2 Binomial lattice

Binomial lattice was proposed by Cox et al. (Cox et al. [1979]) for option pricing. In this approach, the price movement of a stock is constructed as a binary tree with up and down movement of the underlying asset price. That is, at a node representing a particular time, the value of stock can go up by a factor u or go down by a factor of d. u and d were shown by Cox et al. (Cox et al. [1979]) to be evaluated from volatility (σ) using the equations

$$u = e^{\sigma\sqrt{t}} \tag{4.5}$$

$$d = e^{-\sigma\sqrt{t}} \tag{4.6}$$

Let S be the current stock price, the price at the next time step could go up to

$$S_u = S \times u \tag{4.7}$$

or it can go down to

$$S_d = S \times d \tag{4.8}$$

A simple two-step binomial tree is shown in Figure 4.1

To understand the binomial lattice, let us consider a one-step binomial tree for a call option as shown in the Figure 4.2.

Let the initial stock price be S and the strike price be K. S_u and S_d can be calculated using equations 4.7 and 4.8. f_u and f_d represent the option prices (known

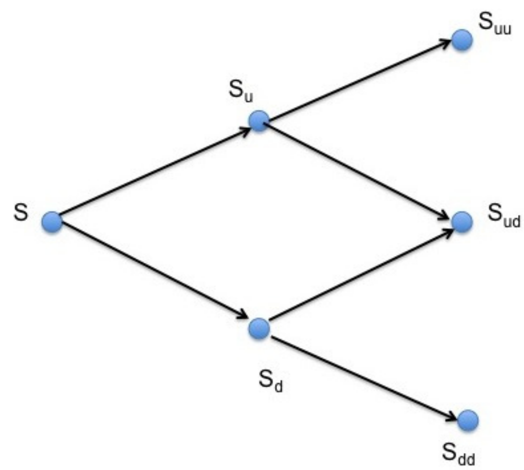

Figure 4.1: Two step binomial lattice

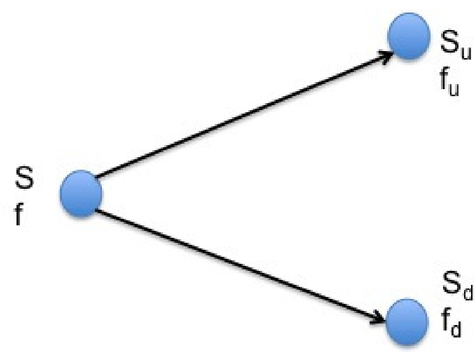

Figure 4.2: 1 step binomial lattice

as pay-off) at the children nodes (leaf nodes in this example) at a future time and f is the option price we want to evaluate at the root node (parent node in this example) that represents current time. f_u can be evaluated using equation (4.9)

$$f_u = Max[S_u - K, 0] \tag{4.9}$$

f_d can be evaluated using equation (4.10)

$$f_d = Max[S_d - K, 0] \tag{4.10}$$

Equations 4.7 and 4.8 are the local pay-off from the option at their respective nodes and retain only positive values of pay-off. If p and $(1-p)$ are the probabilities for the asset price to go up and down respectively, the weighted sum of the future pay-off can be computed from

$$future\ pay - off f = pf_u + (1-p)f_d \tag{4.11}$$

Since this is the value in future, this has to be discounted to find the current value of the option (at the parent node). $e^{-r\Delta t}$ is the discounting factor, multiplying which to the weighted sum in equation (4.11) will give the current value of the option and is presented in equation (4.12). In other words, the option value f at the root node can be calculated using f_u and f_d as

$$f = e^{-r\Delta t}[pf_u + (1-p)f_d] \tag{4.12}$$

where,

$$p = \frac{e^{r\Delta t} - d}{u - d} \tag{4.13}$$

The accuracy of the results increase with the increase in the number of time steps and with a large number of time steps, the results converge to those from Black-Scholes-Merton results. When the step size is small, the discretized capturing of price movement would lead to the continuous solution provided by Black-Scholes-Merton model. The generalized N-step binomial lattice is shown in the Figure 4.3.

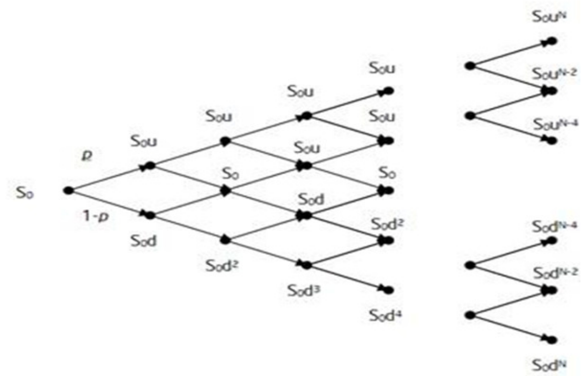

Figure 4.3: N-step binomial lattice

4.3 Monte-Carlo simulations

The use of Monte-Carlo simulations for option pricing was introduced by Phelim Boyle (Boyle [1977]) when he evaluated the European option. Few key research results using this technique were observed in mid 1990s through 2000s. For example, Broadie et al. (Broadie and Glasserman [1996]) evaluated Asian options and Longstaff et al. (Longstaff and Schwartz [2001]) evaluated American options using the Monte-Carlo simulations. Approach taken in these studies and others in finance literature are very mathematical towards closed-form solutions. These approaches cannot be directly applied to price Cloud resources.

The process of evaluating options using Monte-Carlo analysis can be divided into four parts. First, the random walks of the underlying stock are generated. The accuracy of the results improve with the increase in number of such random walks. As shown in Figure 4.4, thousands of random walks are performed for reasonable accuracy. Second, the payoff for each path is calculated i.e. for a call option $S - K$ and for put option $K - S$ are evaluated for all the random walks. Third, these payoffs are averaged and finally the average is discounted to the present day.

This technique is quite suitable for $C3$ pricing as well. A major research challenge in this technique is the generation of random numbers to make a random walk unique from other random walks so that simulations could be effective.

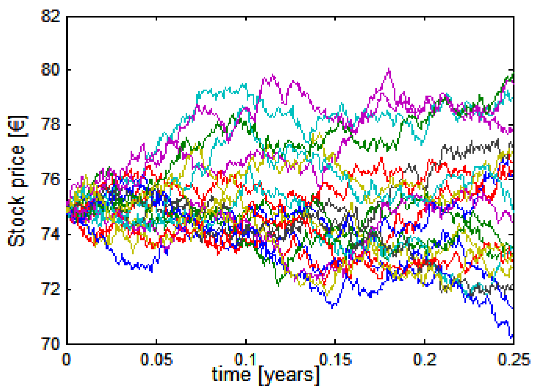

Figure 4.4: Monte-Carlo simulations

Chapter 5

Clabacus

I propose an architecture called Clabacus to price Cloud resources using financial options and present it in this Chapter. Figure 5.1 presents Clabacus (Cloud-Abacus) architecture with various modules that work independently and collaboratively to compute the price of the resources. I explain below each of the modules of this architecture.

Input module: This is the graphical user interface (GUI), where the Cloud resource provider will enter various initial and normal recurring costs; including but not limited to hardware, land lease, energy (electricity, natural gas etc.) cost, high quality personnel costs, and insurance costs. To include other parameters affecting the cost, we have an additional input field for miscellaneous costs.

Input modifier module: This module converts various inputs into one standard unit and one currency. For example, some costs could be in \$/hour (electricity), while some could be in \$/year (land lease).

Driver: This is the main command center of the Clabacus architecture. The prime

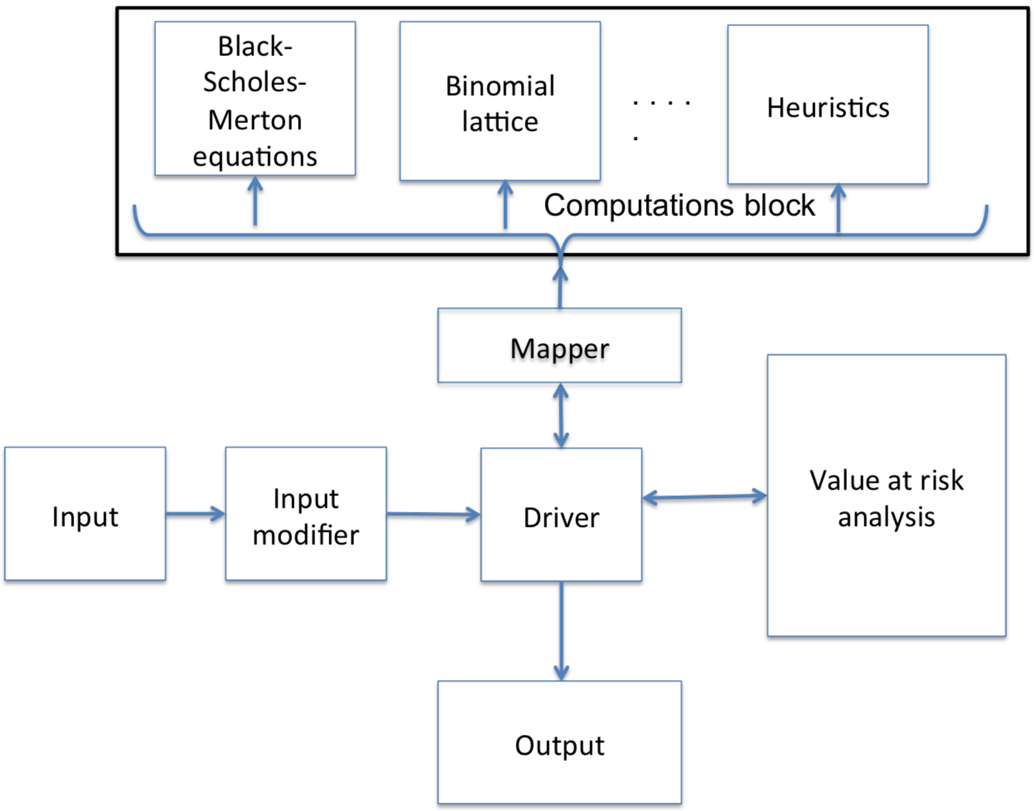

Figure 5.1: The Clabacus Architecture

responsibility of the driver is to select a suitable evaluation (computation) method from among many computational algorithms in the computation block. Based on the user inputs like computation time, desired accuracy and complexities of other user requirements, one particular evaluation method is selected. The second task of the driver is to compute the risk in establishing a data-centre. This is done using the value-at-risk (VaR) module, which is explained further in the next chapter. The VaR module is only used if the user wants to get the price quotes adjusted for risk. The final task of the driver is to assemble the results from computation and VaR modules, before sending it to the output module. Note: While it is expected that accuracy is of utmost importance, sometimes to get an idea on the resource pricing, some clientele would like to get results sooner, which might compromise on the accuracy. Only to accommodate this possibility, I have introduced "desired accuracy" as a parameter in the input module. I present further discussion in Chapter 7.2.

Mapper: Each Algorithm in the computation block uses a varying set of inputs to compute the price of Cloud resources. The task of the mapper module is to fine-tune the input parameters received from driver and feed it to the selected computational algorithm, appropriately.

Computation block: This block includes various computation algorithms (as explained in Chapter 4) such as Black-Scholes-Merton closed form formula for option price, binomial lattice, Monte-Carlo, finite-differencing, fast Fourier transform etc.

Output: This is a graphical user interface and all the outputs will be displayed here.

Chapter 6

Value-at-Risk

Value-at-Risk (VaR) is a measure of potential loss on a specific portfolio. The main uses of VaR are in risk management and financial reporting. Researchers are continuously looking for new and efficient ways to evaluate VaR and the 2008 financial crisis [1] (last accessed May 4, 2016) has given further impetus to finding new and reliable ways of evaluating and using VaR. In this research, I use genetic algorithm (GA) to evaluate VaR for a Cloud Service Provider and compare the results with conventional VaR techniques. In essence, I propose two modifications to the standard GA: 1) normalized population selection and 2) strict population selection. For a typical set of simulations, 8 chromosomes were used each with 8 stored values and we get 8 values for VaR. My experiments show that with increasing mean value of the stock prices, the VaR increases as well. This is not observed in another common approach, the Monte-Carlo simulation, which is experimented in this study as well. Experiments using data from four different market indices show that by adjusting

[1] see for example: http://useconomy.about.com/od/criticalssues/f/What-Is-the-Global-Financial-Crisis-of-2008.htm

the volatility, the VaR computed using GA is more conservative as compared to those computed using Monte-Carlo simulation. The computation cost with GA was found to be smaller compared to Monte-Carlo simulation. Once this new method of computing VaR is tested and compared to the conventional VaR evaluation methods, such as Monte-Carlo simulation, it is then used in pricing Cloud resources. Much of this chapter has been published in an article by Sharma et al. [2015a].

6.1 Value-at-Risk for financial markets

In financial markets, VaR is a measure of risk that determines the potential loss on a specific portfolio. It is an estimate of the maximum loss of a portfolio over a given period of time at a certain confidence level. As mentioned earlier, the primary uses of VaR are in risk management and financial reporting.

Risk management plays a major and cautious role in maintaining financial health of an institution. Risk management can add value to a financial institution by reducing the payable taxes. Income or earnings of a company tend to be very volatile and a sudden increase in the turnover can lead to more taxable income. It is in the best interest of financial planners to smooth the revenue curves, which helps reduce taxes. This is part of risk mitigation, done using VaR.

Another situation where risk management adds value to a company is during financial distress. Financial distress occurs when a company is unable to pay its creditors and is getting closer to declaring bankruptcy. It incurs both direct and indirect costs. Direct costs are due to reorganizations and liquidations. Indirect costs, on the other hand, are associated with the build up of inventory and attrition of highly skilled personnel. In both these cases, proper risk management can aid in reducing financial distress costs. (For more detailed explanation see Cooper [2016])

Risk management can also facilitate optimal investment. Share holders and bond holders of a company are constantly in conflict with each other over the investment decisions of a company. Share holders are more motivated to take higher risks for higher payouts while bond holders are risk averse. Efficient risk management facilitates to ease this conflict for better management decisions.

Managing risk, therefore, is extremely important for proper functioning of an institutions. However, the VaR parameter used to measure risk is not so easy to estimate. If VaR is estimated incorrectly, financial institutions have dire consequences.

In general, VaR measures the potential loss on a portfolio over a certain period of time for a given confidence interval (Damodaran [2016]). That is, as mentioned in Hull [2011], it allows company managers to say that they are X% certain that they will not lose more than V dollars in the next N days.

In simpler terms, VaR is a measure of market exposure on a particular investment. Consider an investment whose pay-offs are normally distributed as shown in Figure 6.1. The VaR with 68.26% confidence level for N business days would be the area under the curve from -1 to 1. Similarly the VaR with 99.73% confidence level for N business days would be area under the curve from -3 to 3. The industry standard for confidence/accuracy level of VaR is usually, 95% and 99%.

We can extend the above example to investments, such as equity call option (Hull [2011]) whose payoff depends on a stock index (S). An increase or decrease in the stock index beyond a particular threshold, known as strike price (K) results in investment

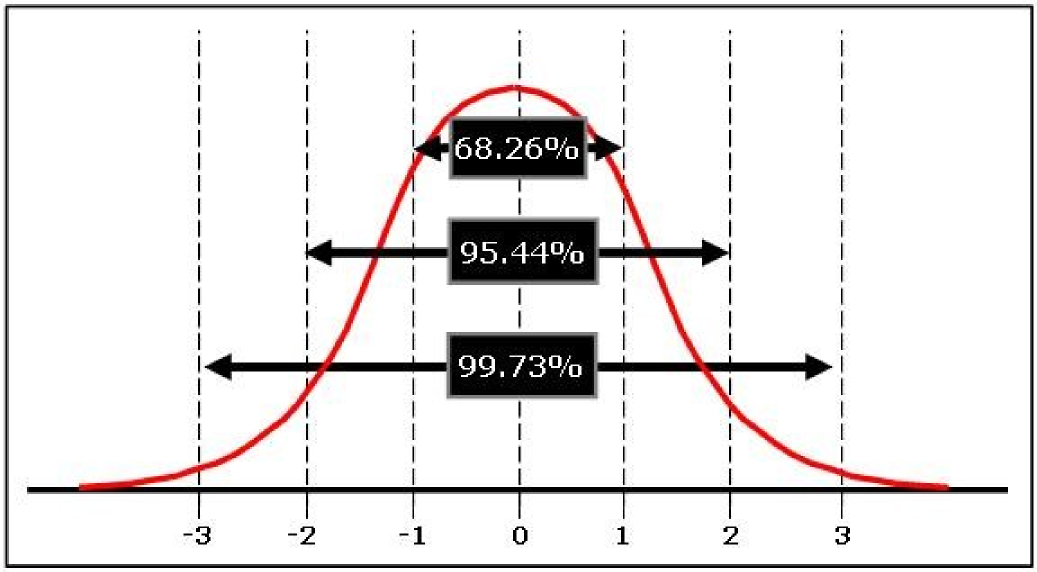

Figure 6.1: VaR for Normal distribution

profit or loss, respectively. The equity call option payoff can be evaluated as given in Equation 6.1 (Hull [2011])

$$\mathcal{P}ay - off \propto (S - K) \tag{6.1}$$

According to Hull [2011] VaR is directly proportional to the payoff,

$$\mathcal{V}aR \propto \mathcal{P}ay - off \tag{6.2}$$

Assume the constant of proportionality in Equation 6.1 is C and is equal to \$1 M. Equation 6.1 can be re-written as

$$\mathcal{V}aR = \mathcal{C}(S - K) \tag{6.3}$$

To evaluate VaR three common methodologies are available: variance-covariance analysis, histogram analysis and Monte-Carlo simulation as explained in Cooper [2016].

6.1.1 Variance-Covariance Analysis

Variance-Covariance (Damodaran [2016], Danielsson and Vries [2000]) also known as analytic or delta-normal method, can be used if the probability distribution of the underlying asset can be evaluated. The delta-normal method is used mostly due to its simplicity. The main hindrance in this approach is the evaluation of probability distributions.

Consider a portfolio with values normally distributed with a mean of \$100 and annual standard deviation of \$10. With 95% confidence we can say that over the next year the value of this portfolio cannot go below \$80 (two standard deviations below mean) and cannot increase above \$120 (two standard deviations above mean). It should be noted that 95% confidence allows the standard deviation of 1.96 on either side of mean and 99% confidence allows the standard deviation of 2.33 on either side of mean (Damodaran [2016]).

This method is used if the portfolio is large and good volatility estimates are available. To use this method the input data should be normally distributed and the data distribution should not have fat-tails (Cooper [2016]).

6.1.2 Histogram simulation

Histogram simulation (Butler and Schachter [1998]) is done by creating a hypothetical time-series of return on a portfolio and it is the simplest way of evaluating VaR. This time-series is created using the historical data (Butler and Schachter [1998]). Statistical tests are done on the historical data to evaluate the VaR. Histogram/historical simulation is used if the data distribution is unknown. This technique is conservative as all the outliers are included to evaluate VaR. Due to the simplicity of this method, even the fat tail distributions can be used as there is no restriction on the data distribution. The limitation of this method is that actual historical data is needed to evaluate VaR, which might not be always available (Butler and Schachter [1998]).

6.1.3 Monte-Carlo Simulation

Monte-Carlo (MC) simulation is used when high precision is required for portfolios. MC is based on the principles of Brownian motion of molecules in fluids (Cooper [2016]). In Brownian motion, the particles of the fluids are assumed to move randomly. MC simulates stock prices over a certain period of time and uses the results of the simulation to measure VaR or the maximum loss over the time period at a confidence level of 95% or 99%. To simulate the stock prices at time, $t + \Delta t$, we use the Equation: 6.4 (Boyle [1977])

$$S(t + \Delta t) = S(t) + \Delta S \tag{6.4}$$

where $S(t)$ is the current stock price and ΔS is the change in stock price which can be calculated using geometric Brownian motion as shown in Equation 6.5 (Boyle [1977])

$$\Delta S = \mu S \Delta t + \sigma S r' \sqrt{\Delta} t \tag{6.5}$$

Equation 6.5 implies that for a given stock priced at \$S with a mean return μ and standard deviation, σ, we can evaluate ΔS for time t. r' is a random number in [0,1].

Once the different values for change in stock price, or ΔS, are evaluated using MC, they are sorted in ascending order and a particular value of ΔS is selected based on the required confidence level. This method consists of many independent simulations and each simulation consists of many cycles. According to Butler and Schachter (Butler and Schachter [1998]) by increasing the number of cycles, we can obtain higher accuracy.

Suppose that we run 100 simulations and get 100 different values of ΔS. These values are sorted in ascending order. If we want 95% confidence level, we select the 95^{th} smallest value and for 99% confidence level, we select 99^{th} smallest value respectively. Two limitations of MC are: (1) large number of simulations are required to get a reasonable value for VaR; (2) the MC assumes normal distribution of the price data, which does not reflect real market scenario. Due to these limitations with MC, I compute the VaR for my research using genetic algorithm.

6.1.4 Genetic Algorithm

Genetic algorithms (GA) (see for example Brabazon and O'Neill [2006]) are a class of nature inspired algorithms, which are formulated using the Charles Darwin's theory of evolution. The theory of evolution is based on the principle of natural selection, which implies the survival of the fittest. The process of natural selection requires a wide range of options to choose from for the best possible offspring. In real world the environment plays a pivotal role in the natural selection. The offspring most likely to survive the current environmental state is chosen to represent the current generation in future. The crossover and mutation (Brabazon and O'Neill [2006]) are the two essential functions that govern the process of natural selection. The crossover function is homologous to reproduction in the real world, where two parents give birth to an offspring. In genetic algorithms the crossover function is used to vary the chromosomes from one generation to the next. The mutation function is used to reflect the changes in the current environmental state. With change in the environmental state, the chromosomes have to adapt to survive in the next generation. The rate of crossovers and mutations define the sensitivity of a particular genetic algorithm.

The initialization phase includes the generation of a random population and evaluating the fitness of each candidate. After this, the process of generating future generations is repeated until the number of iterations is complete or the termination condition is met. During this process two parents are selected and recombined. This process is also known as the crossover. Mutations are also applied to some of the candidates. After this, the fitness of the current population is evaluated and based on the results, the future population is selected.

There are many ways in which the crossover function can be applied. Most frequently used is the single-point crossover function (Brabazon and O'Neill [2006]). Chromosome is the natural vehicle that carries information. Chromosome and parent can be used interchangeably. A chromosome can be visualized as long array of data. In single-point crossover function, two parents are chosen and data after a randomly chosen point is swapped between the parents. This point is just an index of a particular data point. In a two-point crossover function (Brabazon and O'Neill [2006]), the data between two selected points is swapped between the parents. Some other crossover functions are cut-and-splice and uniform crossover functions (Brabazon and O'Neill [2006]). For my research, I use single-point and two-point crossover functions. The algorithm is as follows:

Algorithm 3 Genetic Algorithm

Initialize population with random population

Evaluate each candidate

Repeat until termination condition is met

  Select parents

  Recombine pairs of parents

  Mutate the resulting off springs

  Evaluate new candidates

  Select individual for next generation

  End

End

6.2 Value-at-Risk for Cloud resources

Cloud resource providers or vendors have many clients with different requirements. To accommodate these diverse set of requirements, the vendors have to tailor their service level agreements (SLA). The flexibility in adding or removing clauses in a SLA could determine the number of clients a vendor can have in a multi-tenant environment. Clabacus has the ability to evaluate the risk-adjusted price of a Cloud resource. It is in provider's best interest to accommodate changes to the SLA to facilitate many clients. Clients, on the other hand, also acknowledge the fact that added requirements translate to a higher price. Therefore, it is the client's best interest to understand their business requirements before they request services from providers.

Quality of Service (QoS) is a metric used in SLA to determine the level of services or resources requested. QoS can be thought of as a conglomeration of many parameters, where each parameter governs a single aspect of the service. A standard SLA could have parameters like downtime, maximum number of users supported, storage, memory, CPU cycles and software packages etc. All these parameters when quantized, give a definite value of QoS. Some clients could have a very low tolerance on downtime, while others might have a much lax requirement. Thus, different clients can request different values of elements of a SLA. As the SLA becomes more stringent, the vendor has to quote its prices to encompass all real and notional costs. The real cost includes the electricity charge, cost associated with high skilled personnel and software license fees etc. In general, the real cost is dynamic but identifiable to a large extent and hence can be quantified. It's the notional costs that are hard to evaluate. The notional expenses could include damages caused by fire or natural calamities and the expenses incurred due to SLA violations. In general, the notional expenses can originate from two sources of risk:

1. Risk associated with vendor operations: This is often referred to as the operational risks and are similar to any other kind of manufacturing businesses. Some examples of risks are unfavorable environmental conditions, sudden change in commodity prices resulting in change of electricity prices and events like theft etc. Proper planning and hedging strategies can mitigate most of these risks. An appropriate building design can get rid of most of the risks associated with the unfavorable environment conditions. Hedging in the electricity market can lock in the prices that a vendor has to pay in future and a proper security measures can eliminate cases of theft and burglaries. These risks can be termed as foreseeable risks and should not affect the prices of the Cloud resources. Thus the operational risks are assumed to be very less and not included for my study and hence are not discussed further.

2. Risk associated with SLA liabilities: This risk is associated with the SLA violations. SLA violations can have different consequences on the Cloud providers; from a client not buying any more services to some more serious consequences involving litigations and penalties. When a Cloud provider is unable to provide the promised services, it results in a SLA breach ($\mathcal{P}_{SB}$). All the possible clauses of a SLA breach are also a part of SLA. Cloud providers are very keen to eliminate the SLA liability risk by investing heavily in reliable hardware and software. However, it is highly difficult to eliminate all the risks. All checks and cautions can reduce the risk to bare minimum still leaving a small opportunity for $\mathcal{P}_{SB}$. A SLA breach will have some financial consequences associated with it, so the price quote of a Cloud provider should be adjusted to take into account this inherent risk. VaR estimator is a part of Clabacus and it is used to risk-adjust the price quote. The primary task of VaR estimator is to evaluate the probability of default or breach of a SLA. This probability, represented as percentage, would be added to the initially computed prices ($\mathcal{C}_{base}$) (from computations block) to give a risk-adjusted price.

$$\mathcal{C}_{risk-adjusted} = \mathcal{C}_{base} \times (1 + \mathcal{P}_{SB}) \tag{6.6}$$

$\mathcal{P}_{SB}$ can be evaluated using the QoS. Also, $\mathcal{P}_{SB}$ is equivalent to the confidence level.

Algorithm 4 Risk-adjusted Price for Cloud resources

Get prices from the pricing algorithm 1

For i=1,N (N is the number of iterations)

Evaluate ΔS =f(S,μ, σ)

$C_{risk-adjusted}$= $C_{base-price}$ $\times$ (1+ $\mathcal{P}_{SB}$)

It is the provider's confidence that the SLA will not be breached. Confidence level at the base price gives us the VaR. Therefore, the equation can be rewritten as

$$\mathcal{C}_{risk-adjusted} = \mathcal{C}_{base} + VaR \tag{6.7}$$

It can be seen that with higher VaR, the risk adjusted price of a Cloud resource will increase.

In algorithm 4, μ and σ are the mean and standard deviation of the resource price S. This algorithm provides a generic way to evaluate VaR. The input parameters are resource price, mean, standard deviation and QoS.

The change in the price (ΔS) can be computed using any of the two approaches described below: Monte-Carlo or genetic algorithm.

6.3 Monte-Carlo (MC) Simulation

Once the different values for ΔS are evaluated using MC (as explained in Chapter 6.1.3), they are sorted in ascending order and a particular value of ΔS is selected based on the required confidence level, the VaR. As mentioned earlier, two limitations of MC are: (1) large number of simulations are required to get a reasonable value for VaR; (2) the MC assumes normal distribution of the price data, which does not reflect real market scenario. Due to these limitations with MC, in the current study, we compute the VaR using the genetic algorithm.

6.4 Genetic Algorithm Approach

Another approach in finding the change in price is using genetic algorithm.

6.4.1 Implementation details of GA

A chromosome of length 4 can evaluate 4 different values of ΔS. Consider two chromosomes of length 4 each. Each bit of a chromosome evaluates ΔS and in the fitness evaluation, I average all four bits to find the overall fitness of a chromosome.

(1) *Initialization*: In this step, I initialize all the bits with random initial prices and at the end of this step, chromosomes look like the values in Table 6.1.

Table 6.1: Initial/Crossover/Mutation steps

Initial	ΔS_1	ΔS_2	ΔS_3	ΔS_4
	ΔS_5	ΔS_6	ΔS_7	ΔS_8
Crossover	ΔS_1	ΔS_2	ΔS_5	ΔS_6
	ΔS_3	ΔS_4	ΔS_7	ΔS_8
Mutation	ΔS_1	ΔS_9	ΔS_5	ΔS_6
	ΔS_3	ΔS_4	ΔS_7	ΔS_8

(2) *Crossover*: In this step, the two initial chromosomes (parents) are recombined to form the next generation (Children) of chromosomes as presented in Table 6.1.

(3) *Mutation*: Mutation can take place randomly at any bit and after mutation the chromosomes may look like the one in Table 6.1.

(4) *Evaluate*: At this point all the bits in one chromosome are averaged to get the overall fitness. This overall fitness will dictate the selection of a particular chromosome for next generation.

Once the desired number of simulations is completed, the role of genetic algorithm is completed. At this time all the chromosomes are concatenated to get all the values at the end of last simulation. These values are now sorted and a particular value is selected based on the desired confidence level, which is the VaR. For example, the 95^{th} lowest value selected from the sorted ΔS would mean 95% confidence level.

Various experiments that I designed following the set-up of the driver module, computation block and VaR module of the Clabacus architecture are reported in the next chapter.

Chapter 7

Experiments and Results

I organize this section in the following order: I discuss the Clabacus input parameters first followed by the bounds on the Cloud resource prices that are beneficial for both clients and providers. Then, I discuss the effect of each of the input parameters on the Cloud resource pricing from the client and provider's perspective separately. Much of the results from this chapter are published in an article in IEEE Transactions on Cloud Computing (Sharma et al. [2015b]), while the initial ideas proposed were published in conferences: IEEE/ACM CCGrid(Sharma et al. [2012b]) and IEEE ISPA (Sharma et al. [2012a]).

7.1 Clabacus input parameters

The input parameters to Clabacus are as follows:

Capital investment (I_C): This is the approximate cost of the new equipment or infrastructure the client wants to lease from a Cloud resource provider.

Start time: this is the time when the client starts leasing the Cloud resources.

Use time: this is the duration for which the client actually uses the resources.

Total time: is the time for which the Cloud resource provider will possess a resource. For example, a total time of 2 years means that after 2 years the Cloud provider will dispose the resource from service.

Rate-of-depreciation(θ): is the rate at which the Cloud resource depreciates. This depreciation could be the result of several factors such as the advent of new technology or increased cost of maintenance. One direct effect of this parameter can be seen in the unwillingness of clients to lease a particular (aged) resource.

Quality of service (QOS): is the conglomeration of many factors including completion time, accuracy of results, data confidentiality etc. The quality of service is an input parameter with its value between 0 and 1. As mentioned earlier, 100% quality of service (ie., QoS=1) is preferred by all clients. A Cloud vendor may offer different options to pick from, like memory usage, operating system, processor uptime and security etc. The Client can choose any of these services in any capacity and based on the total usage, QoS will be established. Based on the required Cloud services, a particular QoS can be calculated. Therefore, a QoS of 1.0 can be achieved in many different ways, by selecting a different set of services.

Rate of inflation: is the rate of change of prices on an annual basis.

Price of Cloud resources computed and analyzed with a simple linear regression of above independent variables.

7.2 Lower and Upper Bounds on Resource Prices

This section explains the maximum and minimum cost the service provider would charge a client for leasing a resource.

7.2.1 Upper bound from Compound-Moore's law

The five Cloud variables (parameters) are: capital investment, contract time, rate of depreciation, quality of service and age of resource. Using these together with compound-Moore's law, we can compute the maximum amount the service provider is spending in buying and setting up the hardware resources. In other words, this provides an *upper bound* on the Cloud resource price, the service provider would like to charge a client to recover the investment over the contract period and make profit.

The cost of maintenance (including power, real estate and personnel etc.) is not considered in the computation block for two reasons:

a) My objective is to make Clabacus usable by both users and providers.

The clients would not know apriori the investment costs incurred by the provider and hence they will not be able to include these costs precisely, if they are doing the computations using Clabacus.

b) The revenue generated by providing service to multi-tenants from a given virtualized resource might compensate the maintenance cost incurred by the provider.

Note that this limitation is relaxed in the Value-at-Risk analysis, which takes into account all the recurring costs. Also, risks due to rate of depreciation, rate of inflation, etc. are taken into consideration to adjust the resource prices computed in the computational block. That is, we can expect that adding the cost of maintenance in the price evaluation would bring profits to the provider at an earlier time than normally expected during the life of an infrastructure.

7.2.2 Lower bound from finance models

Using the binomial lattice algorithm or any other techniques (numerical or heuristics) we can compute the minimum Cloud resource price that the service provider would charge a client to recover its initial and recurring investments. In other words, this is the *lower bound* on the Cloud resource price.

7.2.3 Example and Observations

We explain the upper and lower bound concepts presented above through an example. A typical parameter setting for our experiments is: Capital Investment: $300/year; Contract (use) period: 3 years; Rate of depreciation: 10%; Quality of Service (QoS): 0.4; Age of resources (total time): 2 years.

Using compound-Moore's law (Equation 3.3), the upper bound can be calculated as 2.14 cents/hour. This is the maximum amount the service provider would need to charge a client to recover the investment over the contract period. The lower bound using the binomial lattice algorithm is calculated as 1.65 cents/hour. This implies that the service provider would need to charge a client at least 1.65 cents/hour but not necessarily more than 2.14 cents/hour to recover the initial investment.

With the proposed methodology a client is aware of the maximum cost of leasing the resources from the service provider, which is beneficial for the client to compare this cost with respect to buying the resources. In this regard the client would benefit a low cost resource if the price of the leased resource is less than 2.14 cents/hour. Hence, we are reaching an equilibrium condition and since the prices are adjusted continuously for the fluctuation in the market conditions, the resource prices are in dynamic equilibrium.

The equilibrium condition is just a break-even point where the clients and vendors share the profits equally. Proximity of the actual price charged by a vendor would depend on other factors like market competition. In an ideal rational market, the vendor is expected to charge this equilibrium price. The price that the vendor would charge a client should ideally be between these bounds. When the price is between lower bound and equilibrium the vendor would make better profit while the client will be benefiting by leasing the resources at a nominal cost. When the price is between equilibrium and upper bound, vendor would make higher profits while at the same time the client will benefit (a nominal cost considering that clients do not invest in building an infrastructure) based on how close to the bound the actual price of the resource is.

Note that the input parameters mentioned earlier in this section are investments at the time of hardware installation. These parameters, especially the capital investment, need to be adjusted for decreasing prices due to technological evolution as mentioned in Chapter 3. We do this using compound-Moore's law before using these parameters in Algorithm 2 in computing the upper bound. Also, note that the cost to service provider depends on the initial capital investment and remains at 2.14 cents/hour, the upper bound. However, at 100% quality of service (i.e., QoS=1) cost to client is computed as 2.06 cents/hour. That is, the lower bound is 2.06 cents/hour.

In general, if the client had to purchase the resource, the value of the client's infrastructure investment at risk is 100%. However, by leasing the resources from a Cloud service provider the client has none of the investment is at risk of losing value (In fact, there is no prior investment altogether). That is, the VaR for clients is significantly lower.

The example explained above involves one client and one service provider; however in reality a single service provider will be serving many clients simultaneously. With virtualization of the resources, the service provider can cater to the needs of multiple clients from a given physical resource, thus generating more revenue on the initial investment and hence make profit in a so called multi-tenant model.

Based on these observations it can be understood that client and the service provider can have a symbiotic relationship. Experimental results presented in the next subsections with simple linear regression analysis show the influence of various Cloud parameters on the Cloud resources.

7.2.4 Risk-adjusted pricing

I modify the prices computed using any of the algorithms in the computational block to reflect the risks involved (as described in Chapter 6) in providing Cloud services. I present the results from our proposed genetic algorithm, in adjusting the prices for risks.

7.3 Discussion: Pricing from Client's Perspective

In an experiment when studying the effect of one parameter on the resource price, other parametric values are kept constant at a desirable level. In other words, I am doing a simple linear regression analysis as a proof-of-concept of my proposed architecture and pricing model. Also, note that the experimental results in the figures are for a slightly different parameter setting than in Chapter 7.2.3.

7.3.1 Effect of start time on resource price

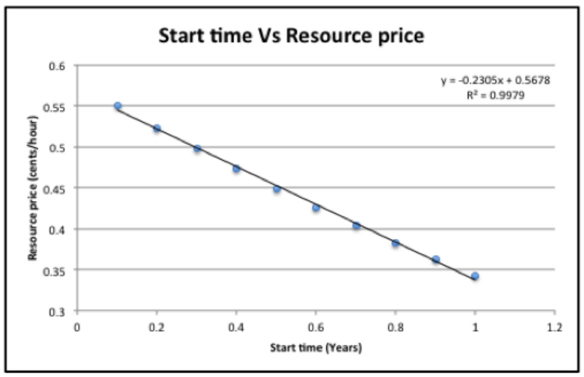

Figure 7.1: Start Time vs Resource Price

Over a period of time the resources the clientele has acquired from the Cloud provider ages, thereby decreasing the value of the resources. Hence, it would be interesting to see the effect of getting computation task done by a resource when the resource is in its mid-life of service. Of course, note that the clientele would not, in general, know the age of a resource. Start time is the time of the client's first instance of using the Cloud resource. As a first experiment, I delay the start time to see the effect on resource pricing and present the results in Figure 7.1. Here, the rate of depreciation is the implicit factor affecting the resource price. This also affects the quality of service obtained from Cloud provider. In other words, using the compound-Moore's law, I show that a client can estimate the rate of depreciation and the effect it has on resource pricing as shown in Figure 7.1. Later in life of a resource, we might expect to pay less for a service from such a resource. This is what we observe in Figure 7.1. However, we may expect that when the pricing is done by the service provider, the age of the resource should not show any effect on the resource price, as long as the quality of service is maintained. This is realized as presented later in Figure 7.11 (Chapter 7.4.5).

7.3.2 Effect of use-time on resource price

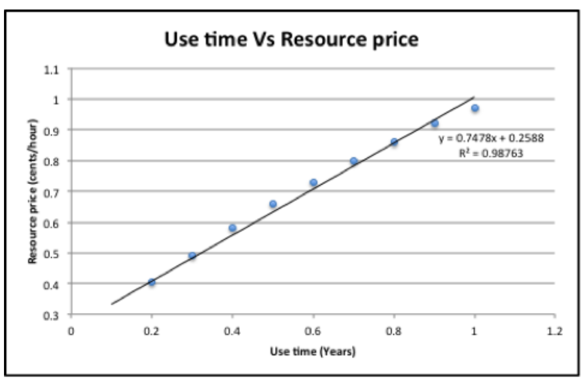

Figure 7.2: Use Time Vs Resource Price

The use-time is the duration of leasing the resources. Effect of this variable is studied and presented in Figure 7.2. Trend in this figure is not only the reflection of expected price rise as use-time increases, rather this is counter-intuitive since, in general, longer a client uses a resource the client would expect to pay lesser. Higher use time on a resource may be interpreted as higher demand on the resource. Note that as the demand of resource increases, the provider will increase the price of the resources. This has to be taken into account by the client as well. This is not easily predictable by a client since the provider may not divulge the information of the load of the resources to the client. However, the computational module in our model has the ability to predict the resource prices over a period of time. For example, in this research, I estimate or predict future prices by emulating many price evolution in the binomial lattice algorithm for a Cloud resource. The resource price computed from binomial lattice is a result of many possible price evolution in the future. In other words, many possible variation in prices are evolved by regenerating the binomial tree for various u, d and σ values to create a many different market conditions and hence appropriate prices using Equation 4.12. Incorporating this financial option based binomial-lattice algorithm (and others in the computational module) allows us to better estimate the price of the resources also, which would not have been possible with simple economic models. This is shown in Figure 7.2.

Note that as already observed in Chapter 5 (mapper module of the Clabacus architecture), this could be related to the accuracy of the pricing results. By refined binomial tree model we might get a better prediction on the prices, while using small number of simulations with MC simulation we might only compute an estimated value of resource price.

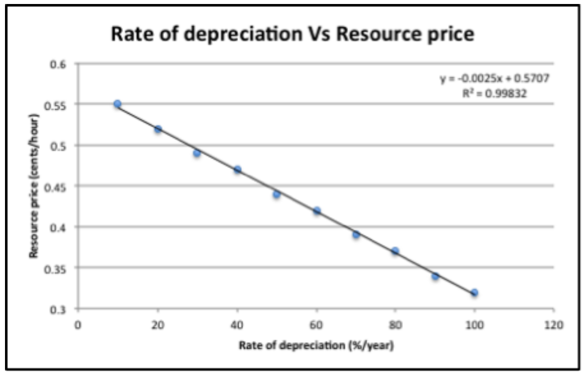

Figure 7.3: Rate of Depreciation vs Resource Price

7.3.3 Effect of rate-of-depreciation on resource price

It is expected that the provide would drop the price on a resource at higher depreciation level of the resources. To study this, a separate regression analysis is made and is presented in Figure 7.3. This figure can be related to Figure 7.1 intuitively. The rate of depreciation is an important factor when leasing a Cloud resource. Recall that in algorithm 1 the rate-of-depreciation is an input parameter to the Compound-Moore's law formula. By finding the strike estimate (which is the resource price) using the binomial lattice algorithm, my model can predict the rate-of-depreciation for the client as shown in Figure 7.3. With a depreciating resource the provider has to incur maintenance costs without which the service charges would decrease as shown in Figure 7.3. Note that in this experiment, other parameters are set to an initial condition, especially the QoS, which cannot be guaranteed without proper maintenance of the infrastructure.

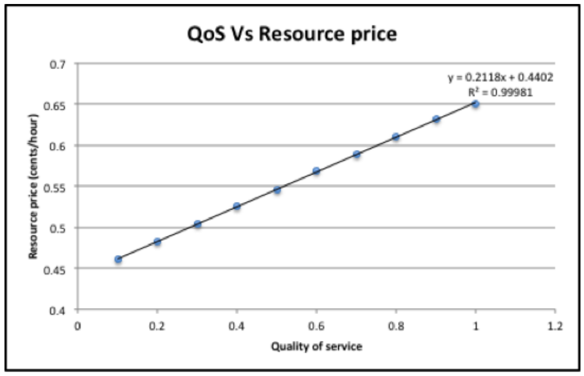

Figure 7.4: Quality of Service vs Resource Price

7.3.4 Effect of quality-of-service on resource price

With the higher QoS (coming out of a resource), it is expected that demand on that resource would increase, thereby, increasing the price of that resource. The binomial lattice model together with the compound-Moore's law captures the resource price (upper bound) for increasing QoS as shown in Figure 7.4. In this figure, my model provides the upper bound price for higher QoS. I can obtain this using my Algorithm 1 where "r" in the binomial lattice algorithm represents QoS. By mapping the appropriate parameters to Equation (4.12), I compute the price of the resources with respect to the QoS.

7.3.5 Effect of rate of inflation on resource price

The inflation rate increases running and maintenance costs of the resources. I can predict the cost of burden on the client due to inflation using my model. Here, I estimate the volatility using the age of resources and rate of depreciation using compound-Moore's law. Also, I calculate the strike estimate using rate of deprecia-

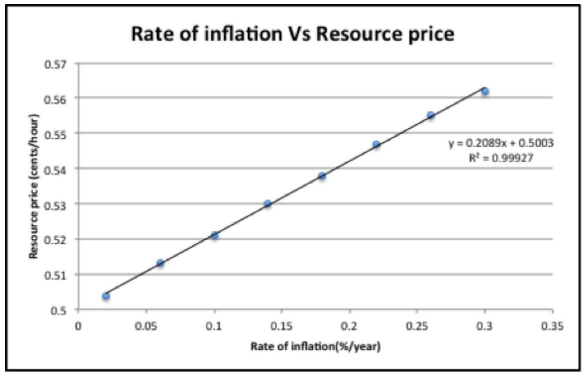

Figure 7.5: Rate of Inflation vs Resource Price

tion and contract time or total time. I map these parameters to the binomial lattice algorithm to obtain the increase in resource prices due to inflation as seen in Figure 7.5. This means that higher resource price(s) is the result of any of the cause(s) such as electricity bill, taxes etc. to the provider.

7.3.6 Effect of capital investment on resource price

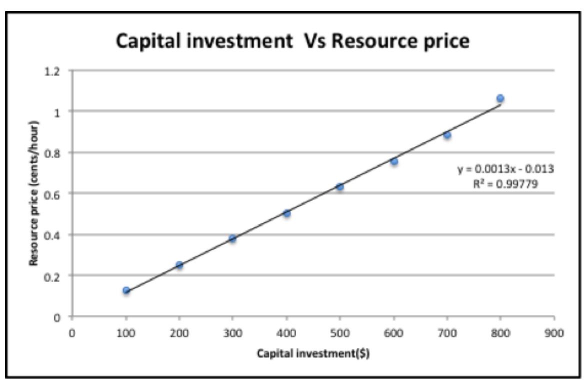

Figure 7.6: Capital Investment vs Resource Price

If a client wants state-of-the-art machine the resource price increases proportionally since the provider would have incurred increasing capital investment. This is observed in Figure 7.6. Capital investment refers to the approximate cost of the Cloud infrastructure that a client would like to acquire instead of leasing from a Cloud resource provider. Given a contract time, initial investment, and the use time of the resources, I can calculate the total investment, strike and volatility from Algorithm 1 using the rate-of-depreciation and age-of-resources. My model uses these parameters to provide the cost of leasing this infrastructure.

7.4 Discussion: Pricing with Provider's perspective

7.4.1 Effect of capital investment on resource price

The effect of increasing capital investment on Cloud resource is presented in Figure 7.7.

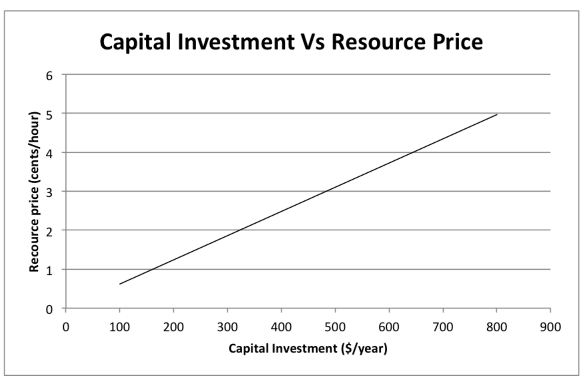

Figure 7.7: Effect of capital investment on the resource price

We see that the resource price (asking price) increase is proportional to the initial investment of the service provider. This proportionality is due to the fact that the contract period is kept constant to avoid the complications due to use-time discussed in Chapter 7.3.2 for linear regression analysis. My algorithm allows me to vary the contract time between a single client and provider. Handling multiple clients with varying contract periods becomes a problem of resource allocation first. Once the tasks are assigned to the appropriate resources, I can price the resources. However, I have not considered the task assignment problem in this study, hence handling multiple use-time possibilities is not considered.

7.4.2 Effect of contract time on resource price

The effect of contract time on the Cloud resource price can be seen in the Figure 7.8.

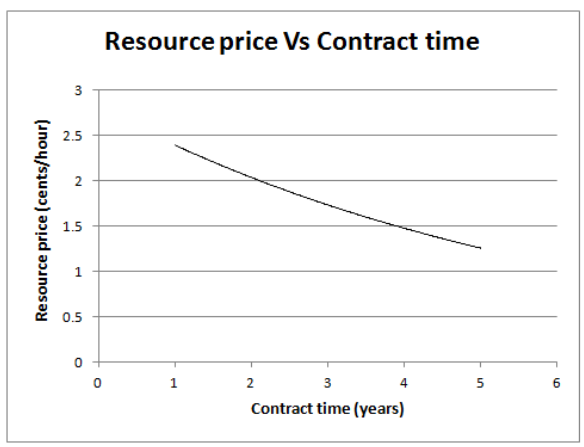

Figure 7.8: Effect of contract period on the resource price

It can be seen that it is beneficial for a client to lease the Cloud resource for a longer time; the prices decrease as the contract time increases as the provider can provide a better pricing for a longer commitment. That is, longer contract periods could benefit from the use of Cloud resources to a larger extent than the smaller contracts. This is due to two reasons: (1) the resource price variation could average out over a long period of time; note that when compared to shorter contracts (minutes) a 2-year contract that we have in our parameter setting can be considered very long period, during which time the price variation is assumed to be normally distributed; and (2) smaller jobs may get executed at a time when the resource price is at its peak, which is still between 2.14 and 1.65 cents per hour (refer to Chapter 7.2.3). Note also that in the current set up, as long as the price is between the lower and upper bounds, both Cloud provider and clients are benefited. It should also be noted that this result is the opposite of the result explained in Chapter 7.3.2. Note that in Chapter 7.3.2, I discuss the client pricing and it is obvious that longer a client leases a resource, s/he has to pay more. In this section, I am examining the price from the vendor's perspective and if the client make a long time commitment, the vendor can offer a more competitive price and hence the price of resource decreases as the contract time increases.

7.4.3 Effect of rate-of-depreciation on resource price

The expected rate-of-depreciation of the hardware installed by the service provider is very critical to price the Cloud resource. As explained earlier in Chapter 7.3.3, if the rate of depreciation is high, the service provider would like to recover its investment before the hardware becomes obsolete, which in turn would increase the price of Cloud resource. This can be seen in Figure 7.9. By cross-referring this figure to Figure 7.3 (Chapter 7.3.3) and Figure 7.10 (Chapter 7.4.4), we can conclude that the Clabacus

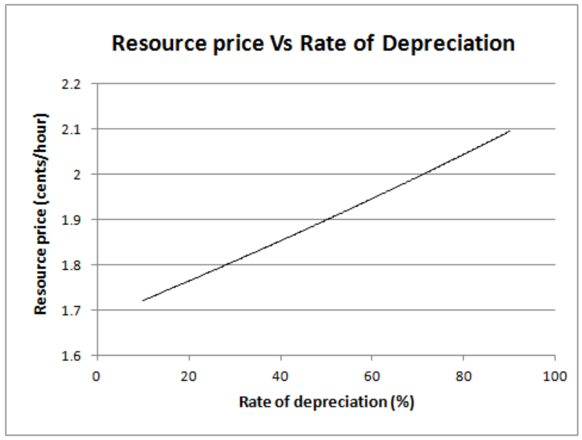

Figure 7.9: Effect of rate of depreciation on the resource price

model brings equilibrium to the providers and clients implicitly.

7.4.4 Effect of quality-of-service on resource price

Higher the QoS the client demands, higher is the expected asking price from the service provider as evident from the Figure 7.10. A price range (lower and upper bounds) presented earlier (in Chapter 7.2.3) is still valid for this discussion. That is, the upper bound price would correspond to the highest QoS. In other words, the compound-Moore's law based pricing and binomial lattice model based pricing form boundaries of the price range for which the QoS varies proportionately. This is consistent with the pricing from the clients perspective as presented in Figure 7.4.

7.4.5 Effect of age-of-resource on resource price

From the provider's perspective, the age-of-the-resource, that is revealed to the clientele should not affect in the resource price as long as the resource provides the

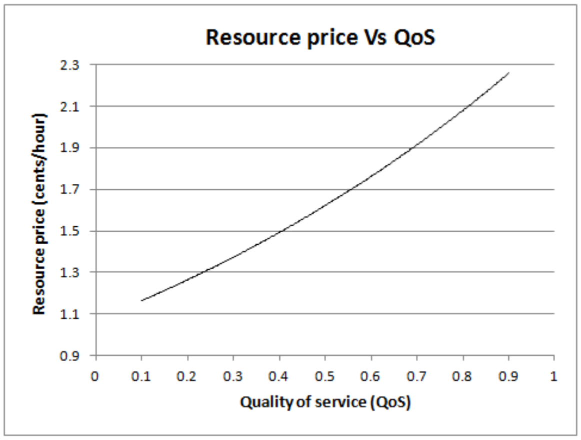

Figure 7.10: Effect of quality of service on the resource price

service without any deterioriating effect on the QoS. The age of resource had no impact on the Cloud resource price as shown in the Figure 7.11. This is because the quality of service and the rate of depreciation are kept constant as we varied the age of resources in our simulations. This implies that the provider is concerned about the quality of service rather than the hardware used to accomplish the task and not to breach the SLA with the set price at the time of contract in completing the task. However, the Cloud service provider might incur more expenses managing aged resources. The clientele is completely immune to it.

7.5 Equilibrium Pricing

Analyzing the blended effect of the parameters on the resource pricing is a natural next step and this is a multi-objective price optimization problem (to be done with multiple regression). Though I have not studied this extensively for my thesis, one simple example is presented in Figure 7.12, where the resource is priced from

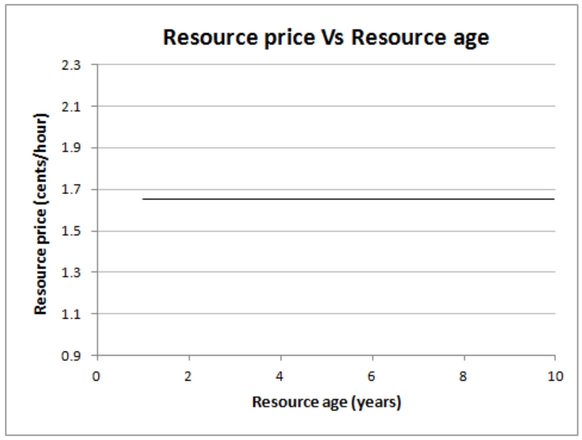

Figure 7.11: Effect of age of the resource on the resource price

both the client and provider perspective for one set of parametric condition different from earlier sections (due to blending). An equilibrium or break-even price of 0.58 cents/hour is obtained at 0.425 years of contract. This result helps a client to make a business decision between buying and leasing. It is beneficial for the client to lease the resource if period of use is below 0.425 years; in other words, the client would exercise the option contract. If the client is going to use the resource for longer than 0.425 years s/he need not exercise the option, that is, s/he need not lease the resources from the provider and instead it would be better to invest in buying the infrastructure. In financial markets, an investor can estimate the premium they expect to pay by computing the option value using any of the pricing models. Recall that the holder of a financial option has the *r*ight to exercise the option. If the market conditions were optimal, the holder of the option would exercise his/her right. The Figure 7.12 signifies this for the Cloud client who is the holder of the contract. Clabacus enables a client to make such a business decision to either buy or lease a resource by knowing an equilibrium price.

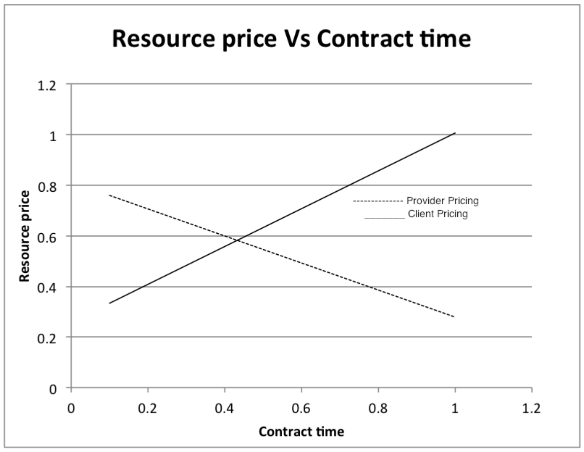

Figure 7.12: Equilibrium price

From this figure, obviously it is not beneficial for the provider to run a data-center for one client with a contract time larger than 0.425 years. However, knowing the overall cost of running the data-center and using the data from this figure the provider can compute the minimum number of clients required to reach a break-even point. In other words, for the Cloud provider the financial impact would be influenced by the number of clients it has among other factors at any given time. The minimum required number of clients can be found using this figure. For example, for a contract time 1.7 years, the provider would need to have 4 clients to reach the break even point of 0.425 years/client.

Referring back to the option pricing models, the writer of an option quotes the premium cost to sell the option based on the market competitors. Similarly, the Cloud provider would base its pricing decisions on the market factors and not only on the equilibrium price computed using the option-based models. That is, in addition to adjusting the resource for the risks involved in setting up the data-centre, the provider would be checking on competitors price as well in deciding on its own price.

The concept of Cloud federation achieves this not only for the resource price but also for resource allocation, task migration, inter-Cloud leasing etc. The model proposed can be extended when pricing is done for a federated Cloud by taking into account this additional constraint of weighing the demand among the members of the federated Clouds, which is left as a future work.

Chapter 8

Conclusions and Future Work

Since the advent of IBM mainframe systems in the 60s, followed by the widespread use of databases in all sorts of industries to the age of personal computers, simpler programs, easy to use computers and cheaper hardware have been driving the research in this field. Grid computing came into prominence in the early 2000s, and its use has been limited to the scientific community and has remained so. Some of the reasons that stopped Grid computing from becoming mainstream were first, high speed internet was not available to the masses, second, the setup, running and maintenance costs were not very different for the mainframes and third, which I feel was the most important reason, the big industrial players did not see much value in it. The lack of interest from the corporate world restricted the use of Grids to scientific communities. Now the times have changed. It has been established beyond doubt that Cloud computing is a revolution in computer science. This impact can be fathomed by the fact that most of the new softwares (including Microsoft Office 365) and mobile applications are Cloud based. Pay-per-use style softwares packages suit perfectly to the clientele and capabilities of Cloud computing. As a consequence of this ongoing revolution, more and more corporations are moving their IT solutions to Cloud. For this migration to happen seamlessly, these corporations need reliable, flexible yet easy to understand Cloud pricing model(s). My thesis contributes to this regime of Cloud computing research.

I presented a quantitative approach to price Cloud resources from both the client and the provider's perspective. As a first step I proposed and designed an architecture called Clabacus (Cloud Abacus). Clabacus has the capability to recognize the input parameters and map them to pricing models appropriately. The challenge in mapping Cloud computing parameters to a financial option model(s) is resolved by carefully analyzing individual parameters and their effect on charges(s) on resource price. I treated the Cloud resources as assets in a finance model to capture the realistic value of the Cloud compute commodities (C3) and used financial option theory concepts and algorithms to solve the mathematical model representing the pricing problem. The finance model provided a lower bound on the prices. The upper bound is found using my proposed compound-Moore's law that takes into account various metrics (such as start time of the resource, use time, rate of depreciation, quality of service, rate of inflation, and capital investment). Risks faced by a service provider or data-center owner is another major issue that affects the pricing of Cloud resources. I have addressed this challenge quantitatively through my genetic algorithm based Value-at-Risk analysis to find the investment risks. The resource price computed in my computation module is adjusted with the VaR to compute the final resource price, and showed that the final price computed is still between the lower and upper bounds

- making the final price competitive to clients and profitable to the provider. The blended effect of the parameters on the resource pricing is a natural next step and this is a multi-objective price optimization problem. I presented one such simple example in Figure 7.12 using financial option pricing model. Nash equilibrium economic principle could be used to achieve such a result, which is an interesting direction to pursue for further research. Through my research, I have brought together the ideas from two different areas: Cloud computing and finance. The marriage of these two areas in pricing the Cloud resources enabled me to use the existing financial models and mapped it to the Cloud resources, hence incorporating various complexities in pricing the Cloud resources. With new Cloud vendors such as Rackspace and Google Cloud, different and more complex financial models need to be used to better price the Cloud resources.

My work caters to the need of scientists and engineers who can go into the details of my novel models, refine and use it according to their needs. My thesis is also for the higher management, who might get some concrete ideas on pricing by studying Figure 7.12. My work can be used in key management meetings between the CEO and the IT team to discuss the cost analysis of Cloud data-center based solutions. As the commercialization of the Cloud technology expands through social networks, I am convinced that, my thesis is a step forward in the right direction at the right time. Moreover, as the use of Cloud data-centre based solutions expands further, more research in Cloud resource pricing, for example, pricing federated Cloud resources would result for which, I strongly believe, my research presented in this thesis would be a forerunner.

List of Publications

1. B. Sharma, R. K. Thulasiram and P. Thulasiraman, Computing Value-at-Risk Using Genetic Algorithm, Journal of Risk Finance, volume 16, pages 170–189, Emerald, 2015.

2. B. Sharma, R. K. Thulasiram, P. Thulasiraman and R. Buyya. Clabacus: A Risk Adjusted Cloud Resources Pricing Model Using Financial Option Theory, IEEE Transactions on Cloud Computing, volume 3, pages 332–344, 2015.

3. B. Sharma, R. K. Thulasiram and P. Thulasiraman, Optimizing Business Processes of Collateral Management System using SQL, 2014 Global Conference on Business and Finance, Honolulu, Hawaii, January 2014.

4. R. Saha, B. Sharma, R. K. Thulasiram and P. Thulasiraman, A Novel Architecture for Financial Investment Services on a Private Cloud, 13th IEEE International Conference on Algorithms and Architectures for Parallel Processing, pages 370–379, Vietri sul Mare, Italy, December 2013.

5. S. Garg, B. Sharma, R. K. Thulasiram, P. Thulasiraman and R. Buyya, Financial Application as a Software Service on Cloud, 5th International Conference on Contemporary Computing, pages 141–151, August 2012, Noida, India.

6. B. Sharma, R. K. Thulasiram and P. Thulasiraman, Portfolio Management Using Particle Swarm Optimization, 10th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), pages 103–110, July 2012, Madrid, Spain.

7. B. Sharma, R. K. Thulasiram, P. Thulasiraman, S. Garg, and R. Buyya, Pricing Cloud Compute Commodities: A Novel Financial Economic Model, 12th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid2012), pages 451–457, Ottawa, Canada, May 2012.

Bibliography

D. Allenotor and R. K. Thulasiram. Grid resources pricing: A novel financial option-based quality of service-profit quasi-static equilibrium model. In *Proceedings (CD-RoM) of the IEEE/ACM Grid 2008*, pages 75–84, Tsukuba, Japan, Sep 2008.

Amazon. http://aws.amazon.com/ec2/ (last accessed on 22 March 2016).

Amazon pricing. http://aws.amazon.com/ec2/pricing/ (last accessed on 22 March 2016).

Y. Amir, B. Awerbuch, and S. Borgstrom. A cost-benefit framework for online management of a meta-computing system. In *1st International Conference on Information and Computational Economy*, pages 155–164, Charleston ,SC, 1998.

A. Anandasivam and M. Premm. Bid price control and dynamic pricing in clouds. In *The 17th European Conference on Information Systems*, pages 328–341, Verona, Italy, June 2009.

M. Armbrust, A. Fox, R. Grifth, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing. *UC Berkeley Reliable Adaptive Distributed Systems Laboratory*, February 2009.

L. Bachelier. The theory of speculation. *PhD thesis, University of Paris*, 1900.

F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, January 1973.

P. Boyle. Options: A Monte Carlo approach. *Journal of Financial Economics*, 4: 223–238, 1977.

A. Brabazon and M. O'Neill. Evolutionary methodologies. *Biological Inspired Algorithms for Financial Modelling*, pages 37–48, 2006.

M. Broadie and P. Glasserman. Estimating security price derivatives using simulation. *Management Science*, 42:269–285, 1996.

J. Butler and B. Schachter. Estimating value-at-risk with a precision measure by combining kernel estimation with historical simulation. *Review of Derivatives Research*, 1(4):371–390, 1998.

R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6):599–616, 2009.

M. Carlos, N. Cook, and S. Shrivastava. On the feasibility of bilaterally agreed accounting of resource consumption. In *International Conference on Service-Oriented Computing (ICSOC 2008)*, pages 270–283, 2011.

P. Carr and D. B. Madan. Option valuation using the fast Fourier transform. *The Journal of Computational Finance*, 2(4):61–73, 1999.

CERN. http://lhc.web.cern.ch/lhc/ (last accessed on 22 March 2016).

P. E. Ceruzzi. *A History of Modern Computing*. The MIT Press, Cambridge, MA, USA, 2003.

C. H. Cooper. *Global Association of Risk Professionals, Financial Risk Manager Exam Study Guide*. GARP, New York, USA, 2016.

J. C. Cox, S. A. Ross, and M. Rubinstein. Options pricing: a simplified approach. *Journal of Financial Economics*, 7:229–263, 1979.

A. Damodaran. http://people.stern.nyu.edu/adamodar/pdfiles/papers/VAR.pdf (undated and unpublished personal website on-line article - last accessed on March 22, 2016), 2016.

J. Danielsson and C. Vries. Value-at-risk and extreme returns. *Annals of Economics and Statistics*, pages 239–270, 2000.

I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. San Franscisco, CA, USA, September 1999.

I. Foster and C. Kesselman. *The Grid2: Blueprint for a New Computing Infrastructure*. 2nd Edition, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, September 2003.

I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud computing and grid computing 360-degree compared. *Grid computing environment workshops*, November 2008.

J. Hull. *Options, Futures and Other Derivates*. Prentice Hall, Princeton, USA, 2011.

A. Hyman. *Charles Babbage: Pioneer of The Computer*. Princeton University Press, Princeton, USA, 1985.

Investopedia. http://www.investopedia.com/terms/c/cloud-computing.asp (last accessed on 22 March 2016).

B. Javadi, R. K. Thulasiram, and R. Buyya. Statistical modeling of spot instance prices in public cloud environments. In *Proceedings of the UCC 2011*, pages 219–228, Melbourne, Australia, December 2011.

S. Kumar, G. Chadha, R. K. Thulasiram, and P. Thulasiraman. Ant colony optimization to price exotic options. In *The IEEE Congress on Evolutionary Computation*, pages 2366–2373, Trondheim, Norway, May 2009.

S. Kumar, R. K. Thulasiram, and P. Thulasiraman. Pricing transmission rights using ant colony optimization. In *Proceedings (CD-RoM) of the ACM GECCO 2011*, Dublin, Ireland, July 2011.

F. Longstaff and E. Schwartz. Valuing american options by simulation: a simple least squares approach. *Review of Financial Studiese*, 14:113–148, 2001.

M. Macas and J. Guitart. A genetic model for pricing in cloud computing markets. In *Proceedings of the 2011 ACM Symposium on Applied Computing (SAC 2011)*, pages 1–6, TaiChung, Taiwan, March 2011.

P. Mell and T. Grance. The nist definition of cloud computing. *National Institute of Standards and Technology, Special Publication 800-145*, September 2011.

R. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.

M. Mihailescu and Y. Teo. Dynamics resource pricing on federated clouds. In *IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pages 513–517, Melbourne, Australia, May 2010.

G. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), April 1965.

Mosso. https://www.mosso.com (last accessed on 22 March 2016).

C. D. Patel and A. J. Shah. Cost model for planning, development and operation of a data center, hp technical report- hpl-2005-107(r.1), 2005.

D. C. Plummer, T. J. Bittman, T. Austin, D. W. Cearley, and D. M. Smith. Cloud computing: Defining and describing an emerging phenomenon. *Gartner inc.*, June 2008.

Rackspace. http://www.rackspace.com/cloud/ (last accessed on 22 March 2016).

M. R. Rahman, Y. Lu, and I. Gupta. Risk aware resource allocation for clouds. *Department of Electrical and Computer Engineering University of Illinois at Urbana-Champaign*, 2011.

Salesforce. https://www.salesforce.com (last accessed on 22 March 2016).

B. Sharma, R. K. Thulasiram, and P. Thulasiraman. Portfolio management using particle swarm optimization on gpu. In *The 10th International Symposium on*

Parallel and Distributed Processing with Applications (ISPA), Leganes, Madrid, Spain, 2012a.

B. Sharma, R. K. Thulasiram, P. Thulasiraman, S. Garg, and R. Buyya. Pricing cloud compute commodities: A novel financial economic model. In *The 12th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid2012)*, pages 451–457, Ottawa, Canada, May 2012b.

B. Sharma, R. K. Thulasiram, and P. Thulasiraman. Computing value-at-risk using genetic algorithm. In *The Journal of Risk Finance*, volume 16, pages 170–189. Emerald, 2015a.

B. Sharma, R. K. Thulasiram, P. Thulasiraman, and R. Buyya. Clabacus: A risk-adjusted cloud resources pricing model using financial option theory. In *IEEE Transactions on Cloud Computing*, volume 3, pages 332–344, 2015b.

R. E. Shrieves and J. M. Wachowicz-Jr. Free cash flow (fcf), economic value added (eva), and net present value (npv): a reconciliation of variations of discounted-cash-flow (dcf) valuation. In *The Engineering Economist: A Journal Devoted to the Problems of Capital Investment*, volume 46. Taylor and Francis, 2001.

S. K. Singh, R. K. Thulasiram, and P. Thulasiraman. A novel application of aco to price transmission rights in electricity markets. In *Proceedings (CD-RoM) of the World Congress on Nature and Biologically Inspired Computing 2013*, pages 267–273, Fargo, ND, USA, July 2013.

D. Tavella and C. Randall. *Pricing Financial Instruments: The Finite Differencing Method*. Wiley, April 2000.

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

M. C. Thomsett. *Options Trading Body of Knowledge, The: Definitive Source for Information About the Options Industry*. FT Press, USA, 2009.

R. K. Thulasiram, P. Thulasiraman, H. Prasain, and G. K. Jha. Nature-inspired soft computing for financial option pricing using high-performance analytics. *Concurrency and Computation: Practice and Experience*, 28(3):707–728, 2014.

A. N. Toosi, R. N. Calheiro, R. K. Thulasiram, and R. Buyya. Resource provisioning policies to increase IaaS provider's profit in a federated cloud environment. In *Proceedings (CD-RoM) of the IEEE HPCC11*, pages 279–287, Banff, AB, Canada, Sep 2011.

E. Walker. The real cost of a cpu hour. *IEEE Computer*, 42(4):35–41, April 2009.

S. Wee. Debunking real-time pricing in cloud computing. In *Proceedings of the CCGrid*, pages 585–590, Newport Beach, CA, May 2011.

M. Woitaszek and H. Tufo. Developing a cloud computing charging model for high performance computing resources. In *10th IEEE International Conference on Computer and Information Technology*, pages 210–217, Madrid, Spain, 2010.