Journal of Theoretical and Applied Information Technology
30th April 2020. Vol.98. No 08

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

COST OPTIMIZATION OF PROCURING CLOUD COMPUTING RESOURCES USING GENETIC ALGORITHMS

[1]RIYADH A.K. MEHDI, [2]MIRNA NACHOUKI

[1]Associate Professor, Ajman University, Department of Information Technology, Ajman, UAE

[2]Assisstant Professor, Ajman University, Department of Information Technology, Ajman, UAE

Email: [1]r.mehdi@ajman.ac.ae, [2]mirna@ajman.ac.ae

ABSTRACT

Cloud computing has given enterprises the opportunity to acquire computing resources cost effectively and yet benefit from other key cloud features that include scalability, instant provisioning, and virtualized resources. Cloud service providers enable businesses to acquire resources by offering different cloud deployment, service, and pricing models. A major challenge for cloud users is to determine the amount of resources to be provisioned that meet their expected needs over the planning horizon, the deployment models to opt for, and the pricing models to adopt to minimize cost. Research in cloud economics has focused on building analytical optimization models that require the representation of the expected demand pattern over the planning horizon as a probability density function amenable to mathematical analysis. In this work, however, we have built a computational model based on simulation and genetic programming to compute the optimal combination of own-private and public cloud resources that satisfy a given pattern of demand as well as the optimal contract guaranteed service level. The model incorporates into the optimization process the different price subscription models offered by cloud providers. The distinguishing features of our model is that it can handle any theoretical or empirical demand probability distribution. In addition, our computational scheme allows for any random variation in any of the parameters affecting the total cost of cloud resources consumed as long as this variation can be described by a theoretical or an empirical density function. The accuracy and correctness of the model was tested against results obtained from mathematical models based on normally and exponentially distributed demand patterns with almost identical results. Thus, our computational model provides a valuable decision tool to help identify the most cost-effective way of provisioning computing resources. Results of experiments conducted in this work indicate that it is more cost effective to use a mixed strategy rather than depend entirely on own-private capacity or on-demand public cloud computing resources alone irrespective of the level of variation in demand; the optimal level of own-private computing capacity is affected by the shape of the demand curve, level of variations in demand, guaranteed service level, and the cloud price subscription model adopted. Future work will extend the computational model to optimize the cost of using cloud storage and networking services. Future work will extend the model to include the cost optimization of using cloud storage and networking services.

Keywords: *Cloud Costing, Cloud Pricing, Optimal Cloud Deployment, Cloud Cost Optimization, Genetic Programming.*

1. INTRODUCTION

The idea of providing computing services as a utility was first proposed by Professor Noah Prywes in the Fall of 1994 when he was delivering an invited speech at Bell Labs. His main point of the talk was a proposal that AT&T should go into the business of providing a computing services to other companies by actually running these companies' data centers [1]. The proposal was to combine data networking with centralized computing centers to provide computing as a service over the Internet [1]. Ten years later, Amazon launched the *Elastic Compute Cloud* (EC2) services which delivered practically what Professor Prywes had envisaged. The EC2 services enabled enterprises located anywhere in the world to create, for a charge, virtual machines in one of Amazon data centers and deploy any software on them. These machines were elastic as the computing resources made available grew with the demand for computing power at appropriate cost and vice versa. Thus, enterprises did not have to invest in acquiring

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and maintaining computing resources, they only pay to the resources and services they are using [1].

Traditionally, acquiring computing infrastructure involved initial investment, operational and maintenance cost of the infrastructure. Developers are often responsible for the design and implementation of the complete system starting from the acquisition of hardware and software up to the implementation of business rules into an application. Applications are run on dedicated infrastructure, and capacity planning was conducted individually for each service [1]. Advances in data networking, distributed processing, and software automation has made cloud computing the most convenient way of acquiring computing infrastructure and services [1]. According to the National Institute of Standards and Technology (NIST), a cloud-computing model has five essential characteristics, three service models, and four deployment models [2].

1.1 Cloud Model Characteristics

The NIST has identified five characteristics of a cloud computing service [2]:

1. On-demand self-service where a customer can provision the resources as needed.
2. Broad network access where clients can access computing resources over the network through standard mechanisms.
3. Resource pooling where physical and virtual resources are dynamically allocated and relocated as needed.
4. Rapid elasticity where resources are provisioned elastically and released scaling upward or downward with demand.
5. Measured services where cloud systems control and optimize resource use by some means of a metering capability.

1.2 Cloud Service Models

A consumer interacts with the cloud through capabilities made available by the cloud service provider. Three main types of models are defined by the NIST, Software-as-a-Service, Platform-as-a-Service, and Infrastructure-as-a-Service [1], [2], [3]:

1. Software-as-a-Service (SaaS) is a solution model in which users use a web browser to access software on demand installed along with programs and user data in the cloud. The customer has no control over the cloud infrastructure apart from some specific user configuration settings. Enterprises that use SaaS eliminate the need for in-house applications, administrative support for the applications, and data storage. Customers using SaaS solutions pay only for the resources they consume. SaaS cloud model provide customers with cost-effective option to get started and an affordable long-term solution. Other advantages include ease of integration and scalability. The disadvantage of SaaS solutions is security. Clients may not trust storing their sensitive data in a remote data-storage facility [1], [3].
2. Platform-as-a-Service (PaaS) models provide a collection of hardware and software resources that developers can use to build and deploy applications within the cloud. This model eliminates the need for customers to buy and maintain hardware as well as the need to install and manage operating systems, databases, and development environments. The resources provided can easily scale up or down according to user needs. The hardware and software within PaaS solution are managed by the platform provider [1], [3].
3. Infrastructure-as-a-Service (IaaS) model provides a virtual data center within the cloud. The cloud provider makes available (physical or virtual) processing, storage networks, and other essential computing resources that enable clients to install and run their software that can include operating systems and other applications. The customer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications. However, the client may have a limited control over the configuration of selected network components [1], [3].

1.3 Cloud Deployment Models

A cloud deployment model specifies how resources within the cloud are used and shared. Each model has its own characteristics with regard to scalability, reliability, security, and cost [1]. The four cloud deployment models defined by the NIST are as follows [1], [2]:

1. Private cloud, computing resources are used exclusively by one entity. The underlying infrastructure can be on or off site. A private cloud offers increased security at a greater cost [1], [3].
2. Public cloud, this model is available for the public and thus less secure. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. Computing resources exists on the premises of the cloud provider. A public cloud is usually the least expensive [1], [3].
3. Community cloud, the cloud infrastructure is provisioned for exclusive use by a specific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

community of consumers that have shared concerns such as security requirements, policy, and compliance considerations. It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises [1], [3].

4. Hybrid cloud, a cloud that consists of two or more distinct infrastructures of private, public, or community clouds that remain unique entities. These clouds are bound together by standardized or proprietary technology that enables data and application portability such as cloud bursting for load balancing between clouds [1], [3].

1.4 Pricing models

Different pricing models of cloud resource provisioning have been suggested. The challenge is to determine an adequate policy of cloud resource subscription that suits cloud customers' requirements. Since resource booking cannot be completed in an adaptive way, traditional resource subscriptions consist of forecasting or assessing the maximum workload to determine the cost. If the number of projected requests is more than the real practice workload, the over-provisioning problem occurs. On the contrary, an under-provisioning problem is shown when the number of projected demands is lesser than the real practice workload [4]. Thus, the provisioning of cloud services comes with a number of different pricing models reflecting different compromises between commitment to providing or using resources and pricing. There are three common pricing models first introduced by Amazon AWS [5]:

1. On-Demand, customers pay-per-use a fixed rate for f the time or quantity of the service used with no commitment on part of the user to the amount of computing resources used.
2. Reserved capacity, with this option the customer commits to a certain amount of use in a given time at a much-reduced cost compared to on-demand option, 60-70%. With this option, cloud providers can plan their capacity more efficiently.
3. On-Spot pricing, this solution allows the customer to bid for unused capacity at 80-90% of the on-demand price but with no commitment from the provider. As soon as the current biding price rises above the customer current bid, the service is withdrawn after the interval for which the resources were hired expires.

1.5 The Need for Cost Optimization

Given the variety of cloud deployment models, different pricing schemes, and the requirements of scalability, flexibility, security, and dynamic workloads make the decision of acquiring cloud resources to meets computing requirements at minimum cost a very complex process. A hybrid cloud model could dynamically allow the customer to adjust the amount of capacity used in a public or private hosting environment thereby achieving high level of scalability and efficiency.

Previous research into the economics of hybrid cloud computing involved the development of theoretical stochastic optimization models where the forecasted demand over the planning period is represented by a parametric probability distribution function under a fixed rate pricing model [6], [7].

This work, however, has three objectives, the first is to develop a computational model based on simulation and genetic algorithms that addresses the optimal mix between privately-owned and public cloud capacity to satisfy a demand pattern described by a given theoretical or an ad hoc empirical distribution function. The second is to further extend the model to determine the optimal mix of public cloud and private resources using a combination of fixed rate and subscription price models offered by cloud providers. Finally, the effect of the guaranteed service level (*gsl*) in a Service Level Agreement (SLA) on the optimal mix of computing resources is investigated.

The performance of our computational model in terms of accuracy and validity has been compared with published results from research work that used mathematical stochastic models in computing the optimal privately-owned computing capacity that can be supplemented by public cloud resources to meet peak demand [6], [7] as we describe later in this work. The remainder of this paper contains a literature review; detailed description and implementation of the suggested computational model; results obtained from test cases, comparisons with published theoretical models, outline of some open research issues, and finally the conclusions and future work.

2. REVIEW OF RELATED WORK

Zaho et al. [8] investigated the problem of minimizing resource rental planning in a cloud environment. The optimization model is based on rental cost analysis of running elastic applications in cloud. Considering the cost tradeoff between data generation and storage, they developed a deterministic optimization model that minimizes the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

unit rental cost of covering customer demand over a planning horizon. They reported that model works well with deterministic cost parameters but not suitable for the spot instance market in cloud computing. By analyzing the predictability of spot price in Amazon EC2, they showed that the spot instance price cannot be well approximated to be used in the deterministic model. This observation had led them to design a stochastic optimization model that seeks to minimize the expected resource rental cost given the presence of spot price uncertainty. Using empirical spot price data sets and realistic cost parameters, they showed that the deterministic model achieves as much as 50% cost reduction compared to the no-planning scheme. And that the stochastic model consistently outperforms its deterministic counterpart in terms of cost saving.

W. Lee et al. [9] have proposed a two-phase approach to define an appropriate procedure for acquiring cloud resources. In the first phase, they developed a mathematical model to compute an upper bound for the optimal amount of long term reserved resources. In the second phase they used Hidden Markov Model to predicate demand for computing resources and allocate virtual machines adaptively based on on-demand provision strategy. Using real-world resource demand data, they indicated that their approach reduced the cost of cloud resources subscriptions significantly.

F. J. Clemente-Castello et al. [10] have developed a cost model specific to running iterative MapReduce applications in a cloud bursting based computing environment. The main issue of hybrid cloud bursting is that the network link between the on premise and the off-premise computational resources often exhibit high latency and low throughput compared to the links within the same data-center. Using this cost model, users can discover trends that can be leveraged to reason about how to balance performance, accuracy and cost such that it optimizes their requirements. Results show that keeping the data on premise in a default storage configuration leads to poor results due to constant remote I/O accesses that stress the weak link. The cost-effectiveness of specialized data strategies stabilizes after very few iterations and greatly outperforms the default configuration. Furthermore, picking the right combination of complementary data-locality strategies has an impact on cost: rack-local asynchronous rebalancing combined with locality enforced scheduling is up to 37% cheaper compared with blocking rebalancing.

S. Chaisiri et al. [11] proposed an optimal cloud resource provisioning (OCRP) algorithm to optimize the total cost of acquiring computing resources by reducing the on-demand cost and oversubscribed cost of under provisioning and over provisioning. The decision model was formulated and solved as a stochastic integer-programming problem with multistage recourse [12] based on uncertain consumer demand and price volatility of acquiring cloud resources. They have also applied Benders decomposition approach [13] to divide an OCRP problem into sub problems which can be solved in parallel as well as the Sample-Average Approach (SAA) [14] for solving the OCRP problem with multiple of scenarios. They indicated that the performance evaluation of the OCRP algorithm has shown that the algorithm can optimally adjust the tradeoff between reservation of resources and allocation of on-demand resources.

Khanafer et al. [15] have developed a cost optimization scheme based on a constrained version of the Ski-rental problem that allows a cloud user to decide whether to rent or buy infrastructure to meet computing requirements. The scheme assumes that the algorithm designer knows the first or second moment of the query arrivals distribution. They reported that the scheme leads to significant cost savings when applied to cloud file systems. However, the scheme does not address the problem of a mixed strategy of provisioning computing resources.

Li et al. [16] have investigated the problem of optimizing both the server running cost and the software storage cost in cloud gaming. They have analyzed the behavior of a proposed stochastic model based on queuing theory under different request dispatching policies. Several classes of computationally efficient heuristic algorithms were experimentally evaluated by simulations with real world parameters. They determined that their proposed Ordered and Genetic algorithms perform quite well in most cases and are robust to dynamic changes. Guo et al. [17] have developed Seagull, a cloud bursting system that determines which applications can run most efficiently on the cloud when local resources are insufficient and move them into the cloud at the appropriate time. Seagull uses a greedy heuristic with an optimization algorithm to optimize the bursting of applications. The system uses selective precopying mechanism to proactively replicate some applications from the private computing resources to the cloud to reduce the migration time of large applications by orders of magnitude. They reported that Seagull has a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

reasonable performance in minimizing cost compared to an Integer Linear Programming solution and its scalability is much better.

Deniziak et al. [18] presented a methodology based on developmental genetic programming for mapping real-time cloud applications onto an IaaS cloud. The aim of the methodology is to find the mapping giving minimal cost of IaaS services required for running the real-time applications in the cloud environment while keeping the level of quality of service as high as possible. Cost reduction of IaaS services is achieved by efficient resource sharing among cloud applications. Henneberger in [6] investigated the economics of hybrid cloud computing. He has developed a simplified stochastic optimization model to identify the conditions under which hybrid cloud computing becomes economically feasible. He stated that under certain conditions it is viable to use cloud services to cover peak demand, even if the price is high or if service levels are low. Furthermore, he indicated that higher variance of demand for capacity should not automatically result into a more extensive use of cloud services. However, his model is not a closed form solution as stated in [7].

Lee [7] has developed a closed form mathematical model to investigate the problem of determining an optimal mix of hybrid cloud computing for enterprise. The model is used to derive a mathematical formula to determine the private capacity that minimizes the total cost of meeting a computing demand described by an exponential probability distribution over the planning period. The author also uses the mathematical model to derive the optimum level of public cloud to be negotiated in a service level agreement (SLA). The shortcoming of the model is that it does not allow for variations in the other parameters that influence the hybrid cloud decision problem such as the price of public cloud computing resources. Moreover, demand for computing resources over a planning period may not follow standard probability distribution amenable to the required mathematical analysis to derive the decision formulae. S. Deniziak et al. [18] have also applied a genetic programming concept to develop an efficient algorithm that finds, in the cloud environment, the minimal cost required for running applications and maintaining the highest QoS. This algorithm schedules and allocates new resources in an optimization way consisting mainly of sharing resources between cloud applications.

3. COMPUTATIONAL MODEL

3.1 Genetic Algorithms

Genetic algorithms are a class of evolutionary search algorithms that solves optimization problems by searching the solution space using a fitness function that evaluates each candidate solution. In contrast to other optimization techniques, an important advantage of evolutionary algorithms is they can deal with multi-modal functions avoiding local optima [19]. Genetic algorithms are loosely modeled on processes that appear to be at work in biological evolution and the working of the immune system [20]. Central to the evolutionary system is the idea of a population of phenotype (chromosomes) that are elements of a high dimensional search space. A phenotype can be thought of as an arrangement of genes, where each gene takes on values from a suitably defined domain of values. In this work, the value is the amount of private capacity that is needed to satisfy an expected level of demand over the planning period where extra demand levels are met through the hiring of public cloud resources. The business objective is to minimize the cost of meeting demand through a mix of private and public cloud resources. Thus, each chromosome encodes one candidate solution for the level of private capacity.

A genetic algorithm starts with a population of randomly generated individuals representing initial possible candidate solutions. The size of such population n is problem and computing resources dependent. Once an initial population has been created, an evolutionary algorithm enters a loop. During each iteration, a certain number of stochastic operators are applied to the current population and a new set of candidate solutions is created that replaces the current one. Each single iteration is referred to as generation as the whole population is replaced by a new better one. Figure 1 shows a basic genetic algorithm iteration. Selection is the first operator to be applied. The aim is to simulate the Darwinian law of "survival of the fittest". In order to create a new transitional population of n individuals, pairs of chromosomes acting as parents are chosen based on their fitness scores. Individual chromosomes are selected for the mating process according to their fitness value. Subsets of a set of random numbers are allocated to individual chromosomes in proportion to their fitness values. Therefore, above average individuals are expected to have more copies in the new population, while below average individuals will risk extinction. When creating new population by crossover and mutation, there exists a distinct possibility that the best chromosome will be destroyed by crossover. To

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

avoid this, the few best chromosomes are first copied to the new population. Crossover and mutation is applied to the remaining chromosomes. The process can increase significantly the convergence of the genetic algorithm [19].

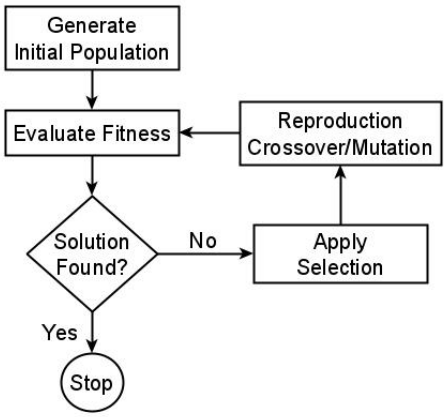

Figure 1. Basic genetic algorithm iteration.

For the computational model developed in this paper, the fitness value represents the overall cost of acquiring private and public cloud resources to meet the total demand for computing resources. To create offspring from the selected parents, a crossover operation is applied. There are many ways of applying the crossover operation. The simplest one is to randomly select a crossover point and swap the genes of the two parents up to the crossover point. Crossover operators may use more than one crossover point to exchange genes. After crossover, the offspring is subjected to a mutation operation. Mutation changes randomly the offspring. The purpose of mutation is to simulate the effect of transcription errors that can happen with a very low probability when a chromosome is duplicated and introduce diversity. This is accomplished by replacing each gene value by another from the domain of possible values using a very low probability of change. The above process is stopped when a termination condition is specified. For example, a predetermined number of generations have been reached, a satisfactory solution has been found or no improvement in the solution quality has taken place for a pre-determined number of generations [19].

3.2 Problem Representation

An enterprise needs to determine the mix of investment in private (own or contracted private cloud) computing resources and those that can be acquired from the public cloud to meet its needs for computing resources. This decision depends on the forecasted demand for the planning period and cost parameters. The decision variables and parameters used in the model are:

Decision Variables:

- private capacity

Parameters:

- *f(x)*: Demand probability density function.
- *pr*: Unit price of own-private resources
- *pb*: Unit price of public cloud
- *gsl*: Guaranteed service level
- *pen*: Cost of unsatisfied demand
- *t*: time periods of the planning horizon.

The following assumptions are made:

Private resources are available at the start of the decision horizon. Public cloud resources can be obtained to satisfy demand that exceed private capacity at a fixed cost. The probability distribution can be theoretical or empirical estimated from historical data. Demand can be divided between private and public cloud resources. The unit price of public cloud over the decision horizon either remain constant or the probability density function of its variation is known. Most of these assumptions are based on Henneberger [6].

The fitness function used to compute the minimum cost of a given private capacity level is outlined below:

```
cloudFitnessFunction(privateCapacity) {
  ▪ Initialize parameters (privateUnitPrice, publicUnitPrice, gServiceLevel, timePeriods, penaltyUnitCost)
  ▪ Initialize parameters of demand probability density function.
  ▪ For each time period t:
      ▪ generate random demand for period t based on the demand cumulative probability density function.
      ▪ if(demand > privateCapacity) {
          ▪ Compute shortage.
          ▪ Compute cumulative Public cloud resources Cost
          ▪ Compute cumulative penalty Cost
      }
  }
 ▪ Compute totalCost as sum of (Private Cloud, publicCost, penaltyCost)
}
```

4. TEST CASES & RESULTS

4.1 Parametric Normal Demand Distributions

To evaluate the performance of our model with respect to the mathematical model developed by Henneberger, the same parameters values were used

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[6]. Thus, we consider the case where stochastic demand follows a normal distribution with a mean of 12 servers and standard deviation of four servers. Cost per server of own-private capacity is $1000, price of public cloud computing resources is $0.14 per hour per server with 99.95% guaranteed service level availability. Penalty cost of not meeting demand is $100 per server-hour, and the number of time units is 8760 (24x365) hours.

The optimal own-private capacity computed from the model is 11 servers with a total cost of $15,062 of meeting demand compared with 10.98 servers and a corresponding cost of $14,569 estimated by the analytical model. Table 1 compares the total cost of different strategies of meeting demand using the analytical approach and our computational model.

Table 1: Comparison Of Model Results With The Analytical Approach.

Provisioning Strategy	Henneberger's Analytical Model	Genetic Based Model
Optimal mix capacity	$14,569	$15,062
Public Cloud Capacity	$19,968	$20,742
Own-private Capacity	$25,539	$25,000

Table 1 shows the cost of different strategies used in provisioning computing resources to meet a given demand pattern using the analytical and the computational models. The results indicate that our computational model has the accuracy of the analytical model, yet it has the advantage of coping with any analytical or empirical demand pattern that can be estimated from historical data. Figure 2 shows a comparison of public cloud unit cost, private cloud unit cost, and overall unit cost of meeting computing demand. For the hypothetical demand used in this study, it is advantageous to use a mixed optimal strategy rather than depend entirely on private own capacity or on-demand public cloud computing resources. Figure 2 also shows that depending on private own resources requires the availability of 27 servers.

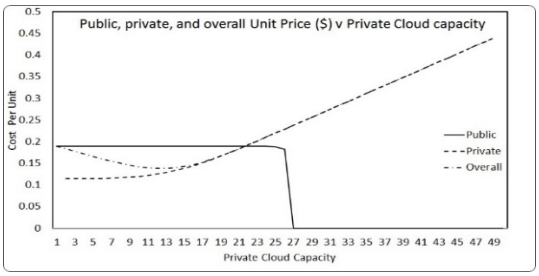

Figure 2. Composition Of The Total Optimal Unit Cost In Relation To Different Own-Private Capacity Levels.

Figure 3 shows the total cost of deploying three strategies: using own-private resources only, on-demand cloud resources only, and a mixed strategy with regard to different level of variation in demand. For any level of variation, the optimal mixed strategy is the most cost effective for the demand pattern under consideration.

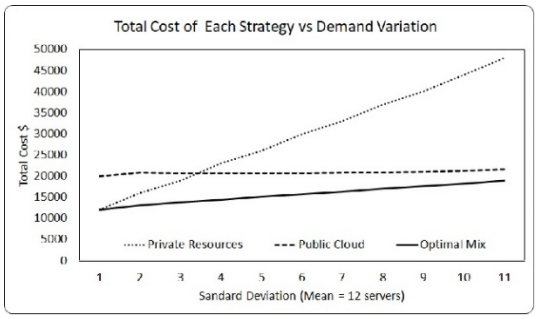

Figure 3. Comparison Of Different Cloud Unit Costs In Relation To Private Capacity Levels.

4.2 Parametric Exponential Demand Pattern

To further test and compare the performance of the model with other analytical approaches, we have used the parameters used by Lee [7] to compare the performance of our model with the performance of his analytical model based on exponential demand density function. The values of the parameters used are: $\lambda(mean\ demand) = 0.001$, pr = $10,000, pen = $100, t = 10,000, pb = $1.0, and gsl = 99.45%. The optimal capacity computed from our model is 439.4, which is very close to the value obtained from Lee's analytical model (434.7). Table 2 compares the results obtained with those of Lee's analytical approach for other parameters.

Figure 4 shows a comparison of the cost per unit of using public cloud resources, own-private resources, and overall unit cost of meeting the demand over the planning period. For the hypothetical demand used in this study (λ=0.001), private own-capacity can be increased to approximately 950 units and the cost is still cheaper than depending entirely on public cloud services. At the optimal own-private capacity level, the cost per unit of public cloud, own resources, and overall cost are 1.54$, 1.23$, and 1.42$ respectively.

Table 2: Comparison Of Model Output With Analytical Approach.

Computed Statistics	Lee's Analytical Model	Genetic Based Model
Optimal capacity	434.7	439.4
Unit cost of public cloud	$1.54	$1.544
Total Cost	$14,331,557	$14,387,446
Capacity Utilization	81.1%	81.26%

Journal of Theoretical and Applied Information Technology
30th April 2020. Vol.98. No 08

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

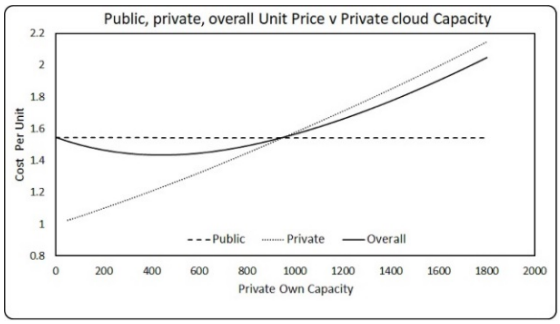

Figure 4. Comparison of different cloud unit costs in relation to private capacity levels.

4.3 Effect of Demand Variation Levels

To examine the effect of demand variation on the optimal level of private capacity and other parameters, we have conducted a number of experiments assuming a normally distributed demand with mean of 24 and a range of standard deviations from a fixed demand (*std* = 0) to a standard deviation of 16. Figure 5 shows that the level of optimal own-private capacity decreases as demand variations increases.

Figure 6 describes the relationship between variations in demand and the overall cost per server-hour using the corresponding optimal own-private capacity. Cost per unit capacity increases as variation in demand increases for the same mean demand. This is to be expected as the excess demand can only be satisfied up to a level determined by the *Service Level Agreement* and the cost of unsatisfied demand is rather high, 100$ per server-hour.

We have also found that the utilization of private optimal capacity decreases as the level of variation in demand increases for a particular demand pattern with the same mean, see Figure 7. This indicates that a high level of variations in demand forces the client to acquire more capacity to avoid the heavy cost of not meeting excess demand.

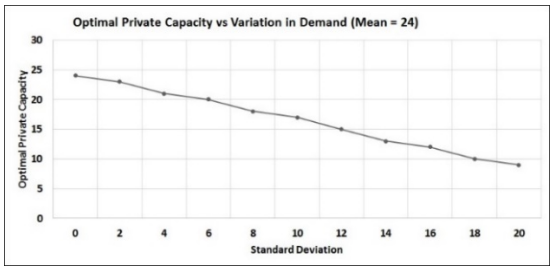

Figure 5. Relationship Between Level Of Variation In Demand And Optimal Private Capacity.

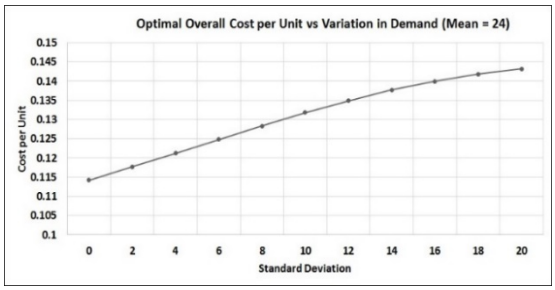

Figure 6. Relationship Between Level Of Variation In Demand And Cost Per Unit.

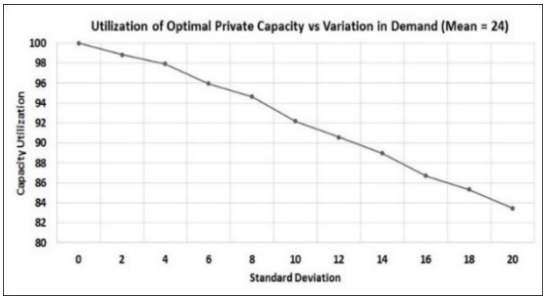

Figure 7. Relationship Between Level Of Variation In Demand And Optimal Private Capacity Utilization.

4.4 Ad hoc Demand Patterns

We have also investigated the behaviour of the model with respect to irregular demand patterns that can not be approximated by a parametric probability distributions. For this purpose, we have synthesised a demand pattern with a mean demand of 24 servers, see Figure 8. The computed optimal private capacity is 19 servers. Figure 9 illustrates the cost per unit of using public cloud resources, own-private resources, and overall unit cost as a function of private own capacity for a hybrid strategy. Figure 9 indicates that the optimal cost of meeting demand through a mixed strategy is 0.158$ per server-hour. In comparison, the cost of meeting demand depending exclusively on own capacity requires 50 servers at a cost of 0.237$ per server-hour and the cost of meeting all demand using on-demand strategy alone is 0.190$.

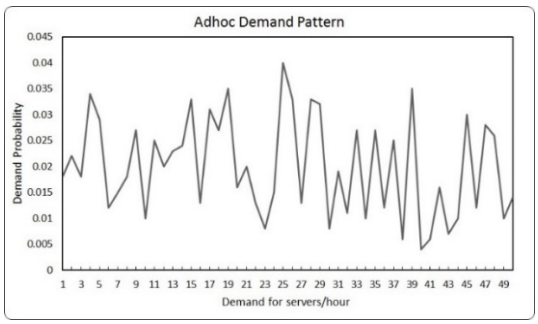

Figure 8. Adhoc Demand Pattern With Mean Of 24.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

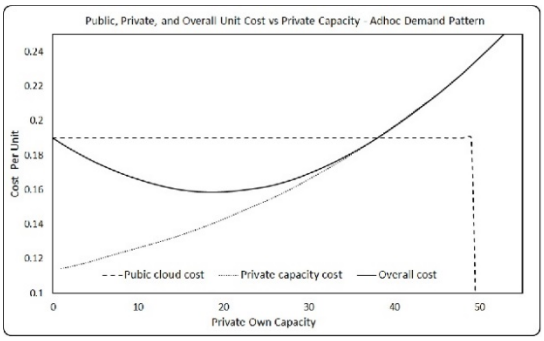

Figure 9. Cost Of Publicly And Privately Met Demand And Total Overall Cost.

4.5 Using a mixture of price models

In the previous analysis, we considered the option of supplementing privately owned capacity by on demand (pay-per-use) cloud resources to meet excessive demand. However, cloud providers have different subscription options to suit the varying requirements of clients as described earlier. To investigate how the optimal cost behaves using a combination of price models, we have extended our model to compute the optimal mix of own-private capacity, and capacity acquired through a reserved instance price model. We assumed that demand in excess of own and reserved capacity is met through an on demand subscription. To run the model, we have used actual pricing data from Amazon Web Services (AWS) with two options: one-year, and three-year reserved instances as shown in Table 3. The model was run for the two reserved instance subscription options, 1-year and 3-year, assuming a normally distributed demand pattern with mean of 10 and standard deviation of 4. Results are summarized in Table 4.

Results indicate that with a reserved instance subscription model, the savings in the overall unit cost of meeting demand is 2.42% of the corresponding cost for pay-per-use option for a 1-year subscription. The corresponding savings for a 3-year subscription is 32.34% due to the price structure of reserved instances shown in Table 3. Thus, for long-term operations, it pays to acquire computing resources through reserved instances options and meeting capacity shotages through the on-demand option.

Table 3: AWS Pricing Data For Amazon EC2 Reserved Instances.

Subscription Option	Upfront Payment	Effective Hourly Rate per Server	On-Demand Hourly Rate
1-Year	$501	$0.057	$0.096
3-Year	$968	$0.037	

Table 4: Summary Of Results For The 1-Year And 3-Year Subscription Options (Mean 12, Std 4).

Optimal Capacity, Cost and Utilization Statistics	Subscription Option		On-Demand Option
	1-Year	3-Year	
Optimal own-private capacity (Servers)	0	0	9
Optimal subscription capacity (Servers)	11	14	-
Own-private capacity unit cost ($)	-	-	0.1198
Subscription unit cost ($)	0.1197	0.0816	-
On-demand unit cost ($)	0.1460	0.1460	0.1460
Overall unit cost ($)	**0.1249**	**0.0866**	**0.1280**

We have also experimented with two demand senarios to explore the effect of higher variations in demand on the cost of acquring computing resources throuhg own-private capacity, reserved instances, and on-demand subscription. In the first senario, we assumed a normally distributed demand with mean of 24 and standard deviation of 4. In the second senario, we changed the standard deviation to 8. Results are shown in Tables 5 and 6.

Comparing Table 5 with Table 6, it can seen that for higher and more stable demand (*mean* 24, *std* 4), the optimal capacity is acquired through a mix of own-private computing resources (12 servers) and reserved instances (11 servers) compared with only reserved instance (11) for the case were the *mean* is 12 and *std* of 4.

Comparing results in Tables 6 and 7, we can observe that the overall price per unit has increased by 4.6% for 1-year reserved instances, 8.1% for 3-year reserved instances, and by 5.78% for the on-demand subscription only. However, it is still more cost effective to use a reserved instances provision strategy rather than to rely on own capacity supplemented by on-demand subscription.

Table 5: Summary Of Results For The 1-Year And 3-Year Subscription Options (Mean 24, Std 4).

Optimal Capacity, Cost and Utilization Statistics	Subscription Option		On-Demand Option
	1-Year	3-Year	
Optimal own-private capacity (Servers)	12	0	21
Optimal subscription capacity (Servers)	11	26	-
Own-private capacity unit cost ($)	0.1142	-	0.1166
Subscription unit cost ($)	0.1197	0.0777	-
On-demand unit cost ($)	0.1460	0.1460	0.1460
Overall unit cost ($)	**0.1196**	**0.0803**	**0.1212**

Table 6: Summary Of Results For The 1-Year And 3-Year Subscription Options (Mean 24, Std 8).

Optimal Capacity, Cost and Utilization Statistics	Subscription Option		On-Demand Option
	1-Year	3-Year	
Optimal own-private capacity (Servers)	0	0	18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Optimal subscription capacity (Servers)	21	28	-
Own-private capacity unit cost ($)	-	-	0.1205
Subscription unit cost ($)	0.1195	0.0823	-
On-demand unit cost ($)	0.1460	0.1460	0.1460
Overall unit cost ($)	0.1251	0.0868	0.1282

4.6 Optimal Public Cloud Guaranteed Service Level

A service-level agreement defines the guaranteed service level to the customer. The higher the guaranteed service level, the higher the price. To minimize the overall cost of acquiring computing resources to meet demand, a customer needs to find the optimal level of guaranteed service level. Lee, in [7] has used the formula shown below to describe the relationship between the public cloud (on-demand) cost and the guaranteed service level:

$$p = base\_price + (gsl - base\_level)*psr, \text{ -------- (1)}$$

where p is the unit cost of public cloud, *base_price* is the unit cost of public cloud for a base level guarantee, gsl is the required service level guarantee by customer, *base_level* is the base level guarantee offered by the provider, and *psr* is the premium service rate at which higher levels of guaranteed service levels above the base level are offered [7].

The computational model was extended to so that the *gsl* parameter becomes a decision variable. The genetic algorithm optimization function is modified to find the optimal combination of *gsl*, and own-private cloud capacity that minimizes the total cost using an on-demand provisioning strategy only. The results obtained below are based on a base service level of 99.90% and a base level price of .096$/unit. The optimal *gsl* was computed for a range of values for the the premium service rate, *psr*. Figure 10 describe the relationshp between the optimal guaranteed service level and the premium service rate. The graph indicates that when the premium rate increases, there comes a point where the increase in the cost of on-demand resources due to a higher level of *gsl* outweigh the cost of not meeting demand at that *gsl*. Thus, it becomes more cost effective to incure the penalty of not meeting demand by contracting for lower a *gsl*. In our experiment, for a given demand pattern, if the *psr* is above 1$, then contracting cloud resources at 99.90% is more cost effective than contracting resources at levels close to 99.99%.

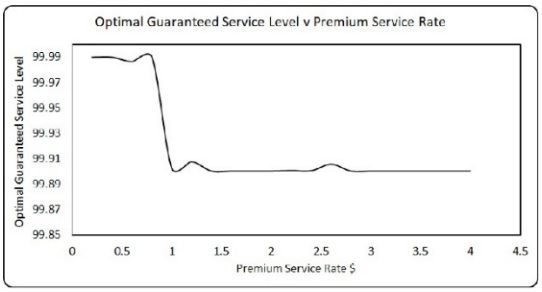

Figure 10. Optimal Guaranteed Service Levels V Premium Service Rates.

Figure 11 shows that as the premium service rate increase, the optimal own-private cloud capacity increases correspondingly and then stabilizes since premium rates above 1$ will result in a *gsl* near the base level and consequently higher premium rates of service levels will not have an impact as the value of the term (*gsl – base_level)* in equation (1) approaches zero keeping the price of on-demand resources constant. On the other hand, as the premium price drops below 1$, required own-private capacity decreases. When the premium price drops to zero, customers contract at the highest guaranteed service level offered by the cloud provide and the private cloud capacity drops to 7 servers. However, when the premium service rate increase above 1$, own-private capacity stabilizes at 11 servers, and excess capacity is met through on-demand subscription subject to the guaranteed service level.

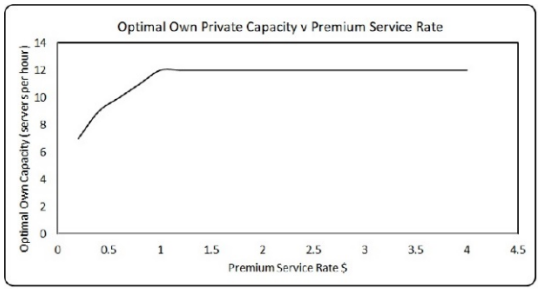

Figure 11. Optimal Private Capacity V Premium Service Rates.

Figure 12 demonstrate the relationshp between the premium service rate and the optimal cost per unit for a given demand pattern. Again, as the premium service rate approaches 1$, the optimal *gsl* drops to the basic level and the term (*gsl-base_level*) drops to zero and consquently, higher levels of *psr* will have no effect on the basic unit price of on-demand public resources.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

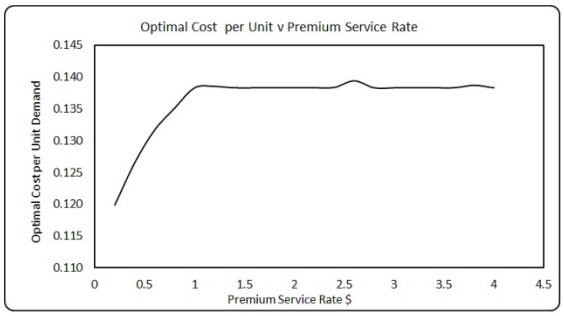

Figure 12. Total Optimal Cost V Premium Service Rates.

5. OPEN RESEARCH ISSUES

Researchers need to address some challenges in the area of cloud cost optimization. In this work, we suggest four research trails that have a direct impact on the cost of acquiring cloud resources. First, cloud providers offer computing resources based mainly on fixed or subscription price models. Researchers should investigate the effect of a dynamic pricing model on the cost of cloud services acquired by customers [21]. Second, very often, the design and build of cloud applications do not take into account the architecture of the cloud deployment platform. Consequently, there is a need to estimate the effect of specific application features on the cost of running the application on that platform. For example, the cost of a particular query for a widget installed in a web application may be expensive to run. Third, the lack of proper provisioning models is another area where research is needed.

A cloud provider should be able to anticipate the computing resources it needs to avoid under-provisioning or over-provisioning. Under-provisioning leads to low performance and high job latency. On the other hand, over-provisioning leads to idle capacity resulting in higher costs to the users. In effect, the customer is paying for the scalability feature provided by the cloud. Fourth, another area of interest to cost optimization is the lack of metrics that allow cloud users to forecast and manage cloud costs. Cloud platforms, like AWS, provide an automatic scaling feature to control cloud cost by adjusting capacity. However, forecasting and controlling cloud costs is sophisticated when business demand for existing resources increases, decreases or fluctuates. Finally, it is also interesting to see how inventory models can be applied to cloud cost optimization from a cloud provider or a customer perspective [22].

6. CONCLUSIONS

Given the variety of cloud deployment models, different pricing schemes, and the requirements of scalability, flexibility, security, and dynamic workloads make the decision of acquiring cloud resources to meets computing requirements at minimum cost a very complex process. A hybrid cloud model using different pricing options offered by cloud providers could dynamically allow the customer to adjust the amount of capacity used in a public or private hosting environment thereby achieving high level of scalability and efficiency. Research in this area has focused on building analytical models that require the representation of the demand pattern over the planning horizon as a mathematical probability density function amenable to mathematical analysis. Another shortcoming of these models is that they use an on-demand fixed-rate pricing model.

In this work, we have built a simulation model based on genetic programming and simulation to determine the optimal combination of own-private and public cloud resources that satisfy a given demand pattern described by a parametric or ad hoc empirical demand distribution constructed from historical data. In addition, the model take into consideration the different pricing options available in determining the optimal mix of public and own-private resource as well as the optimal guaranteed service level. Results obtained from the model are almost identical to those obtained from comparable analytical models in the literature under the same conditions and parameters, confirming the validity and accuracy of our model. Our experiments have shown that the optimal level of own-private computing capacity depends to a large extent on the shape of the demand curve, variation levels in demand, price models, and the guaranteed service level. These conclusions can be summarized as follows:

- it is more cost effective to use a mixed optimal strategy rather than depend entirely on own-private capacity or on-demand public cloud computing resources alone irrespective of the level of variation in demand.
- the proportion of optimal own-private capacity decreases as demand variations increases causing more dependence on public cloud resources.
- The total cost per unit capacity of computing resources increases as variation in demand increases for the same level of average demand.
- for long-term operations, its cheaper to acquire computing resources through reserved instances

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

option. However, with a more stable demand, the optimal capacity is acquired through a mix of own-private resources and reserved instances. In either case, computing capacity is supplemented by on-demand resources to meet demand shotages.

- as the premium service rate increase, the optimal own-private cloud capacity increases correspondingly and then stabilizes when higher premium rates force customers to opt for a guarantteed service level close to the base level. The optimal cost per unit for a given demand pattern behave similarly.

Future work will extend the computational model to optimize the cost of using cloud storage and networking services.

REFERENCES:

[1] I. Faynberg, H. –L. Lu, and D. Skuler, *Cloud Computing: Business Trends and Technologies*. West Sussex: United Kingdom, Wiley, 2016.

[2] P. Mell, and T. Grance, Special Publications 800-145: The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology. US Department of Commerce , Gaithersburg, MD, September 2011.

[3] K. Jamsa, "*Cloud Computing*". Burlington, MA: Jones & Bartlett Learning, 2013.

[4] Michael Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. S., and M. Zaharia, "A view of cloud computing", Communications of the ACM, vol. 53, p. 50, Apr. 2010.

[5] Amazon.com, Inc, Seattle, WA, USA, "How AWS Pricing Works," June 2018 [on line]. Available: hhtp://aws.amazon.com/whitepapers/

[6] M. Henneberger, "Covering peak demand by using cloud services – an economic analysis," *Journal of Decision Systems*, Vol. 25, No. 2, pp. 118-135, 2016.

[7] L. Lee, "Determining an Optimal Mix of Hybrid Cloud Computing for Enterprises," *Companion Proceedings of the 10th International Conference on Utility and Cloud Computing*, UCC 2017, Austin, TX, USA, December 5-8.

[8] H. Zhao, M. Pan, X. Liu, X. Li and Y. Fang, "Optimal Resource Rental Planning for Elastic Applications in Cloud Market", IEEE 26th International Parallel and Distributed Processing Symposium, pp. 808 - 819, 2012.

[9] W-R. Lee, H-Y. Teng, and R-H. Hwang, "Optimization of Cloud Resource Subscription Policy", IEEE 4th International Conference on Cloud Computing Technology and Science, pp. 449 - 455, 2012.

[10] F. J. Clemente-Castello, R. Mayo, and J. C. Fernandez, "Cost Model and Analysis of Iterative MapReduce Applications for Hybrid Cloud Bursting", 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 858 – 864, 2017.

[11] S. Chaisiri, B-S. Lee, and D. Niyato, "Optimization of Resource Provisioning Cost in Cloud Computing", IEEE Transactions On Services Computing, Vol. 5, No. 2, pp. 164 – 177, 2012.

[12] F. V. Louveaux, "Stochastic Integer Programming", Handbooks in OR & MS, vol. 10, pp. 213-266, 2003.

[13] A.J. Conejo, E. Castillo, and R. Garcıa-Bertrand, "Linear Programming: Complicating Variables", Decomposition Techniques in Mathematical Programming, chapter 3, pp. 107-139, Springer, 2006.

[14] J. Linderoth, A. Shapiro, and S. Wright, "The Empirical Behavior of Sampling Methods for Stochastic Programming", Ann. Operational Research, vol. 142, no. 1, pp. 215-241, 2006.

[15] A. Khanafer, M. Kodialam, and K. Puttaswamy, "To Rent or to Buy in the Presence of Statistical Information: The Constrained Ski-Rental Problem," *IEEE/ACM Transactions on Networking*, Vol. 23, No. 4, pp. 1067-1077, 2015.

[16] Y. Li, Y. DENG, X. Tang, W. Cai, X. Liu, and G. Wang, "Cost-Efficient Server Provisioning for Cloud Computing," *ACM Transactions on Multimedia Computing, Communications and Applications*, Vol. 14, No. 3s, Article 55, 2018.

[17] T. Guo, U. Sharma, P. Shenoy, T. Wood, and S. Sahu, "Cost-Aware Cloud Bursting for Enterprise Applications," ACM Transactions on Internet Technology, Vol. 13, No. 3, Article 10, 2014.

[18] S. Deniziak, L. Ciopinski, G. Pawinski, K. Wieczorek, and S. Bak, "Cost Optimization of Real-Time Cloud Applications Using Developmental Genetic Programming," *Procedings of the IEEE/ACM 7th International Conference on Utility and Cloud Computing*, 2014.

[19] N. Nedjah, A. Abraham, and L. Mourelle, "Evolutionary Computation: from Genetic Algorithms to Genetic Programming," in *Genetic Systems Programming: Theory and Experiences*, N. Nedjah, A. Abraham, and L. Mourelle, Eds. Springer, 2005, pp. 2-9.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[20] P. Mars, J. R. Chen, R. Nambiar, *Learning algorithms: theory and applications in signal processing, control and communications*. CRC Press, 2018. [E-book] Available: Taylor & Francis Group.

[21] K. Hamadache, V. Simko, R. Dautov, F. Gonidis, P. Zerva, I. Anaya, and A. Polyviou, "Cost in the Cloud: Rationalization and Research Trails," *The Second International Conference on Advanced Cloud and Big Data*, Huangshan, Anhui, China, November 20-22, 2014.

[22] A. Nodari, "Cost Optimization in Cloud Computing," M.S. Thesis, School of Sc., Aalto University, Otaneimi, Finland, June 2015. Accessed on: Mar. 17, 2020. [Online]. Available: https://aaltodoc.aalto.fi/handle/123456789/17711.