Optimizing Long-term IaaS Service Composition

Sajib Mistry, Athman Bouguettaya, Hai Dong(✉), and A.K. Qin

School of Computer Science and Information Technology,
RMIT University, Melbourne, Australia
{sajib.mistry,athman.bouguettaya,hai.dong,kai.qin}@rmit.edu.au

Abstract. We propose a new economic model based optimization approach to compose an optimal set of infrastructure service requests over a long-term period. The service requests have the features of variable arrival time and dynamic resource and QoS requirements. A new economic model is proposed that incorporates dynamic pricing and operation cost modeling of the service requests. A genetic optimization approach is incorporated in the economic model that generates dynamic global solutions considering the runtime behavior of service requests. Experimental results prove the feasibility of the proposed approach.

1 Introduction

Cloud computing is increasingly becoming the technology of choice as the next-generation platform for conducting business. Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) solutions have already been offered by big companies in the cloud market [1]. An IaaS provider receives service requests in the form of computing resource (i.e., functional) and Quality of Service (QoS) (i.e., non-functional) requirements. The provider associates resource specifications (e.g., CPU, Memory, and Network Bandwidth) and QoS attributes (e.g., availability, throughput, and response time) with provided Virtual Machines (VMs) [2]. For the latter, the provider usually specifies them in a Service Level Agreements (SLA), where SLA violations may incur penalties for the provider [1]. The key challenge for the provider is the optimal composition of the service requests that closely meet the provider's economic expectation by considering certain constraints, such as resource limits and SLA violations.

The provider-consumer relationship between IaaS and SaaS providers is long-lasting and economically driven [3]. The contract period of a long-term service are usually counted in months or years. Due to the nature of the cloud consumers, e.g., multi-tenancy and changing business and cost requirements, a fixed set of requirements are often inapplicable in a long-term period. The key characteristic of the long-term service requests is that its *functional* and *non-functional* requirements change from time to time [3]. The long-term economic expectation of an IaaS provider also can change over time. Here we consider profit, SLA violations, and resource utilization as the key components in the long-term economic expectation. For example, a provider may find that SLA violations in the summer period influence the reputation more than the other periods. In this

A. Barros et al. (Eds.): ICSOC 2015, LNCS 9435, pp. 333–342, 2015.
DOI: 10.1007/978-3-662-48616-0_22

research, we assume that the long-term economic expectation of an IaaS provider has already been defined. The objective is to select an optimal set of long-term service requests that satisfy the provider's long-term economic expectations.

In our previous research, we propose a new model for predicting dynamic consumer request behavior in long-term IaaS service compositions [4]. We observe that SaaS providers' run-time service requests can be different from their initial requests. However, an effective economic model based optimization process using the predicted data is missing in the existing approaches. For example, the under-utilized resources for some service requests can be reallocated to fulfill other suitable service requests to maximize the profit. To the best of our knowledge, existing composition approaches only consider the composition of short-term service requests [5,6]. We identify that two long-term factors, i.e., *economic model* and *dynamic optimization* are missing in the existing approaches. The long-term economic model evaluates a composition of requests using the future predicted economic factors, such as dynamic pricing and operation cost. The dynamic optimization process operates at different times of a composition interval to improve the efficiency of the global composition.

We propose a new long-term IaaS service composition framework that composes stochastic service requests dynamically. We represent the long-term service requests as multidimensional time series. Existing approaches consider the short-term economic model of the IaaS provider [7–9], which is inapplicable for a long-term period. *We propose an economic model for cost and revenue analysis considering the long-term aspects of IaaS provider.* The proposed economic model is constructed as a Dynamic Bayesian Network (DBN) [10]. We use a multidimensional time series matching approach to evaluate a composition's fitness to the given long-term economic expectation. Next, *we propose a genetic algorithm (GA) for long-term service composition using the economic model.* Although combinatorial optimization techniques such as Integer Linear Programming (ILP) and dynamic programming are the preferable choices for service composition [4], they cannot be used in IaaS service combination, as the economic model is non-linear in nature [9]. The proposed approach deals with the non-linear objective function by creating a population of feasible solutions.

2 Related Work

A prediction model for long-term dynamic behavior of the consumers is proposed in [4]. A Mixed Integer Linear Programming(MILP) based optimization process is used with short-term economic and *one-time* arrival model of requests in [4]. Short-term resource allocation algorithms for SaaS providers are proposed in [5] to minimize the infrastructure cost and SLA violations. A task scheduling algorithm is proposed in [6] that uses analytic hierarchy process to optimally allocate resources. Heuristic algorithms are proposed to determine whether a new request can be admitted without impacting accepted requests in [11]. We propose a dynamic IaaS service composition framework that considers stochastic arrival of the requests and long-term economic model of the provider, which is a new area in this field.

The long-term QoS economic model of a service consumer is proposed in [2]. The model states how QoS requirements of end users change over time. An economic model for self-tuned catching is proposed in [7]. An economic model of federated cloud is described in [9]. The model evaluates the cost of using resources from a federated cloud and develops a resource management core for the profit maximization. Existing economic models, nevertheless, do not consider long-term relationships among composite QoSs, resource utilization, and operation costs. Generic penalty-based and repair-based methods for composition are proposed in [12]. However, such methods are not incorporated in long-term economic models for dynamic environments. Hence, we need to develop an economic model based dynamic optimization technique for IaaS service composition.

3 The Long-term Economic Model of the IaaS Provider

We represent the long-term service requests as multidimensional time series. Each long-term service request of the i^{th} consumers is defined as a tuple, $U_i = \{c_t, m_t, nb_t, av_t, rt_t, th_t |\ t\ \epsilon\ [1,T]\}$. Here, $c, m,$ andnb are the required units of CPU, Memory, and Network respectively. The QoS requirements $av, rt,$ and th specify the required units of Availability, Response time, and Throughput respectively. $[1,T]$ is a composition interval. Let us assume that there are k requests (S_k) in the map denoted as $MAP(S_k) = \{< U_1, Arrival_1 >, .., < U_k, Arrival_k >\}$. The requests are already transformed into runtime behavior and the arrival times are predicted in a request map. The task of the economic model is to generate the long-term economic valuation of the request map. We consider profit (P), number of SLA violations (NO_SLV), and resource utilization factor(UF) as the key attributes in the economic valuation. The next task is to measure the closeness of the composition to a given long-term economic expectation.

3.1 The Long-term Economic Valuation

The first task is to convert the request map $MAP(S_k)$ to the long-term economic valuation denoted as $EVAL(t) = \{P_t, NO\_SLV_t, UF_t \mid t\ \epsilon\ T\}$. We use long-term revenue and operation cost modeling to calculate the profit. As the business models of IaaS providers are similar to business models of utility providers, *demand-driven pricing model* is a common phenomena in the cloud market. For example, the price of a EC2 service fluctuates up to 80 % in the Amazon cloud spot market [9]. As Dynamic Bayesian Network (DBN) models succeed in modelling temporal dynamic environments [10], we represent the dynamic pricing behavior as a DBN. The DBN describes the correlations among physical resources (computing (C), storage(M) and network(NB)), QoS values(Availability (AV), Throughput (TH), and Response time (RT)), demand, and service price. The model calculates the probability (Q) of a service price X at t for the service request $U^{(t)} = \{C, M, NB, AV, TH, RT\}$ given its previous price at time $(t-1)$ as follows:

$$Q(X, U^{(t)}, t) = P(Price_t = X | U^{(t)}, Price_{t-1}) \tag{1}$$

The IaaS provider may serve exiting and new requests at the same time. Due to the pricing model, a service may be priced differently for different consumers. As long-term requests reserve the resources at current prices, the price is correlated with the arrival time. For instance, a consumer who arrives at time t always pays the price advertised at time t. As each request in $MAP(S_k)$ is associated with its arrival time, we sort them in an ascending order. Let us denote, $Q(X, Y, U^{(t)}, t_0)$ is the initial probability table for the request U_0. We generate the individual probability table $Q(X, Y, U^{(t)}, t_i)$ for request U_i using Eq. 1. We calculate the revenue time series (REV_t) of $MAP(S_k)$ using the Maximum Probable Explanation (MPE) algorithm [10] on $Q(X, Y, U^{(t)}, t_i)$ in Eq. 2.

$$P_i ... P_0 = arg\ max\ \{Q(X, U^{(t)}, t_i) \mid Q(X, U^{(t)}, t_{i-1}), ...Q(X, U^{(t)}, t_0)\} \tag{2}$$
$$REV_t = \sum_i^{|MAP(S_k)|} P_i, \text{ where } U^{(t)} \epsilon\ U_i$$

We apply the following assumptions to model the long-term cost:

1. **The fixed costs of the provider are distributed over the servers for the amortization period.** Such costs include data center building cost, land cost, hardware price, etc. Each server i has a capital cost in a time unit t represented as $CAPEX_{it}$. Generally t is treated as a month or quarter in a year. If the provider has N physical servers, the fixed cost at time t is calculated by the following equation:

$$Fixed\_Cost_t = CAPEX_{it} \times N. \tag{3}$$

2. **The operation cost is determined by the server power consumption.** Although there are other costs, such as maintenance cost, employee salary, and insurance cost, we choose the power consumption cost for simplicity. ARIMA is a popular model to predict the dynamic price of power [9]. In Eq. 4, the auto regressive (AR) part depends on the p lagged values of the time series of aggregated AG_n where L is the lag operator and α_i is the coefficient constant. The moving average (MA) part depends on the q lagged values of the previous prediction errors (ϵ_t) and θ_i is the coefficient constant. d represents the number of times that the difference operation is performed to obtain the stationary time series. The values of (p, d, q) is determined by the Box-Jenkis method for the price history [4].

$$(1 - \sum_{i=1}^{p} \alpha_i L^i)(1 - L)^d Power_t = (1 + \sum_{i=1}^{q} \theta_i L^i)\epsilon_t. \tag{4}$$

3. **Each running physical server should have a predefined threshold (σ) of utility.** For example, the provider may decide that physical server will not run unless the resources are expected to be used more than 30 %.

The number of physical servers that will run to satisfy requests depends on the resource allocation module installed in the server [9]. The module requires the composed request from individual service requests. The composed request time series ($\{\bar{C}_t, \bar{M}_t, \bar{NB}_t, \bar{AV}_t, \bar{TH}_t, \bar{RT}_t\}$) for $\hat{MAP}(S_k)$ can be formed using the composition rule of resources [5] and composition rules of the QoSs [3] as follows:

$$\text{Resource Composition: } \bar{X}_t = \sum_i^{|\hat{MAP}(S_k)|} X_i^{(t)} \text{ where, } X = \{C, M, NB\} \quad (5)$$

$$\text{QoS Composition: } \bar{AV}_t = max(av_i^{(t)}); \bar{TH}_t = max(th_i^{(t)}); \bar{RT}_t = \sum_i^{|\hat{MAP}(S_k)|}(rt_i^{(t)}); \text{ where } i \epsilon \hat{MAP}(S_k)$$

Our long-term economic model is independent of any particular resource allocation schemes. We assume that the resource allocation scheme installed in the physical servers has a function (F) to convert the composed requests to an utility factor (UF) as:

$$UF_t = F(\bar{C}_t, \bar{M}_t, \bar{NB}_t, \bar{AV}_t, \bar{TH}_t, \bar{RT}_t) \quad (6)$$

Let us assume that a server has a maximum utility factor UF_{max}. Hence, $\lceil \frac{UF_t}{UF_{max}} \rceil$ of the physical server is needed to satisfy the composite requests. Each running physical server has two types of power cost units: (a) fixed power cost for the routine operation of the server $(fixed\_unit)$ and (b) variable power cost $(variable\_unit)$ to satisfy the utility factor. If UF is linearly proportional to the $(variable\_unit)$ with the constant Var_{cons}, we can calculate the operation cost at time t using the following equation:

$$OP\_Cost_t = Power_t \times (fixed\_unit \times \lceil \frac{UF_t}{UF_{max}} \rceil + Var_{cons} \times UF_t) \quad (7)$$

The proposed framework allows SLA violations to occur. The SLA violation cost $SLA\_Cost(t)$ depends on the SLA violation penalty rate $(SLA\_Penalty)$ and the number of SLA violations(NO_SLV_t) in a certain time t. We calculate the number of SLA violations and the constrained satisfied utility factor using Algorithm 1. Algorithm 1 checks whether a composition satisfies all the constraints. It reduces the number of service requests until all the constraints are satisfied. The SLA violation cost (SLA_Cost) is calculated using the following equation:

$$SLA\_Cost_t = SLA\_Penalty \times NO\_SLV_t \quad (8)$$

The revenue, the operation cost, and the SLA violation cost are used to calculate the profit of a composition. Algorithm 1 determines the number of SLA violation and the resource utilization factor. We generate long-term economic valuation of the request map $MAP(S_k)$ using Eqs. 2, 3, 7, and 8 as follow:

$$EVAL(t) = \{REV_t - Fixed\_Cost_t - OP\_Cost_t - SLA\_Cost_t, NO\_SLV_t, UF_t | \ t \epsilon T\}. \quad (9)$$

3.2 Long-term Economic Expectation and Fitness of a Composition

We include profit, number of SLA violation, and resource utility factor as the key attributes in the long-term economic expectation. However, the influence of the attributes may not be equal all the time. For example, SLA violations at peak hours may be more important than the profit when considering business reputation. We incorporate such influence weights in the multidimensional time series, $EXP(t) = \{(P_t^E, W_t^P), (NO\_SLV_t^E, W_t^S), (UF_t^E, W_t^{UF}) | \ t \epsilon T\}$.We define the fitness of a composition as the distance between the long-term economic valuation of

Algorithm 1. Determining the number of SLA violation and Utility Factor of a composition

Input: Request Map: $MAP(S_k)$, Resource and QoS constraints:($C_{max}, M_{max}, NB_{mx}, AV_{max}$, RT_{max}, TH_{max}), server utility threshold σ and maximum utility factor UF_{max}.
Output: the number of SLA violations (NO_SLV_t) and Utility Factor (UF_t) at time t

NO_SLV_t:=-1
repeat
  NO_SLV_t:=NO_SLV_t+1
  Set $Candidate(|MAP(S_k)|) = \{i \; \epsilon \; MAP(S_k)\}$
  Generate $X_t := (\bar{C}_t, \bar{M}_t, \bar{NB}_t, \bar{AV}_t, \bar{TH}_t, \bar{RT}_t)$ using $Candidate(|MAP(S_k)|)$ in Eq. 5
  Generate $UF_t := F(Y_t)$ using Eq. 6
  Server utility threshold $\hat{\sigma}$:=(UF_{max} mod UF_t)
  Remove any requests from the $MAP(S_k)$ with a random distribution and update $|MAP(S_k)| := |MAP(S_k)| - 1$
until $X_t < X_{max} \mid X \; \epsilon \; \{C, M, NB, AV, RT, TH\}$ and $\hat{\sigma} < \sigma$
Return NO_SLV_t and UF_t

the composition and the economic expectation. For simplicity, we deploy the Euclidean Distance as the default distance function. As smaller distance infers better fitness, we formulate the weighted fitness function in Eq. 10. This fitness function acts as the objective function in the proposed genetic optimization.

$$Fitness(MAP(S_k)) = DIS(EXP, EVAL) = \sqrt{\sum_{t=1}^{T} \frac{W_t^P}{(P_t^E - P_t)^2} + \frac{W_t^S}{(NO\_SLV_t^E - NO\_SLV_t)^2} + \frac{W_t^{UF}}{(UF_t^E - UF_t)^2}}. \quad (10)$$

3.3 Genetic Optimization Using the Economic Model

The proposed economic model is non-linear in nature. We design a genetic algorithm (GA) to address the non-linear properties in IaaS economic model. The task of the optimization process is to find an optimal subset $\{MAP(S_k) \mid k < N\}$ that maximizes the fitness of the composition stated in Eq. 10. The GA simulates evolutionary processes by considering an initial *population* of *individuals* and applying genetic operators in each reproduction. In optimization terms, each individual in the population is encoded into a string or *chromosome* that represents a possible solution to a given problem. We use a binary string representation $X[j] = 0$ or 1 to represent the possible solution of $MAP(S_k)$. The j^{th} request is considered in the solution if $X[j] = 1$. For example, a candidate solution $\{B, C\}$ is represented as $\{0110\}$ in the IaaS composition if the incoming requests are $\{A, B, C, D\}$. The fitness of an individual is evaluated based on the fitness function in Eq. 10. Highly fit solutions reproduce new "offspring" solutions (i.e., children) by exchanging pieces of their genetic information in a *crossover* procedure. Mutation is often applied after crossover by altering some genes in the strings. The less fit chromosomes in the population are replaced by the new children. This process is repeated until a satisfactory solution is found. We use a binary tournament selection method [12] to generate the initial population for the first optimization. Crossover point is set in the middle point of chromosome. We set the mutation rate as 2 bits per child. After discarding the duplicated child in crossover operation, the new population replaces the individual chromosomes with the lowest fitness value (steady-state replacement). We continue generating new populations until the solutions are not improved.

The requests in the long-term IaaS composition are based on their predicted future behavior. However, existing requests in the system may behave differently than predicted. We use fixed interval (ΔT) to check whether the existing composition is deviating from the prediction. Other check points are the predicted incoming arrival times of new requests. For example, assume that $\{(A, t_0), (C, t_5)\}$ is the initial IaaS composition. If $\Delta t = 2$, the optimization check points are t_3 and t_5. If request B arrives at time t_2, we may consider inclusion of B if the new predicted behavior of A is different from its initial prediction. Hence, the optimization process should be run again from scratch to update the solution.

4 Experiments and Results

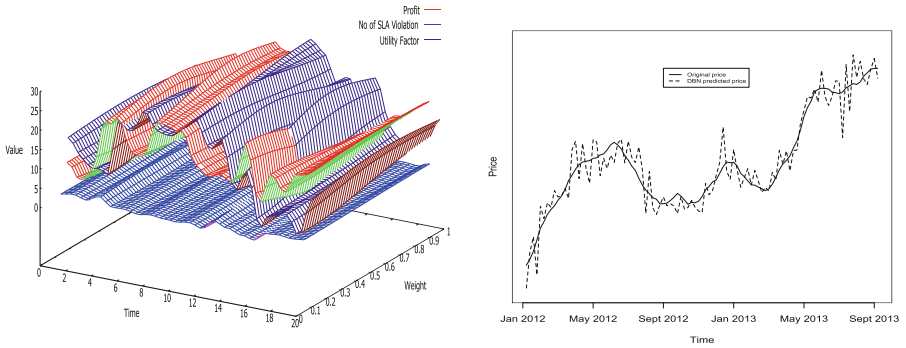
A set of experiments are conducted to evaluate the efficiency of the proposed long-term economic model and genetic algorithm. At first we evaluate the prediction accuracy of the economic model. The fitness of the economic model based composition is compared with a greedy and brute-force approach. In the greedy approach, the provider does not consider the future arrival and valuation of the requests. It accepts the requests that does not violate the SLA at the time of arrival [11]. The brute-force approach considers all the possible compositions to choose the optimal composition. All the experiments are conducted on computers with Intel Core i7 CPU (2.13 GHz and 4 GB RAM). R statistical tool [13] is used to implement the algorithms. We evaluate the proposed method using a mixture of Google Cluster resource utilization [14], real world cloud QoS performance [15], and synthetic data. We randomly generates 100 arrival time points and attach them with the requests to generate a stochastic map of 100 long-term requests. We synthetically generate the runtime behavior of the service requests using Correlation Density Index (CDI) [4]. A higher CDI refers to a lower randomness of the correlations between the service request and the runtime service usage in the history. We generate five sets of service runtime requests with 5 different CDI values (0.5, 0.6, 0.7, 0.8, 0.9) from the map of 100 long-term requests. The long-term economic expectation is also synthetically created (Fig. 1(a)). The mean and standard deviation of the attributes in economic expectation are as follows: profit (mean: \$100, sd: \$20, weight: 0.5), number of SLA violation (mean: \$3, sd: \$2, and weight: 0.3), and utility factor (mean: 80 %, sd: 30 %, and weight: 0.2).

4.1 Setup of the Long-term Economic Model

The proposed economic model incorporates the change of service price and operation cost. We use the time series of electricity price (E_t) from Australian Bureau of Statistics [16] to generate the change in the service price and operation cost. The price of resources and QoS are set by following a weighted Rackspace pricing model as (\$5 $\times$ E_t/unit per hour for any types of resources). [9] provides a short-term mapping relationship between operation cost and utilization. We

Table 1. Long-term relationship between Utility Factor and Operation Cost

UF	Cost/hour	UF	Cost/hour	UF	Cost/hour
5 %	$\$110 \times E_t$	20 %	$\$120 \times E_t$	30 %	$\$140 \times E_t$
60 %	$\$150 \times E_t$	90 %	$\$160 \times E_t$	100 %	$\$165 \times E_t$



(a) (b)

Fig. 1. (a) Long-term Economic Expectation of the provider (b) Prediction accuracy of DBN

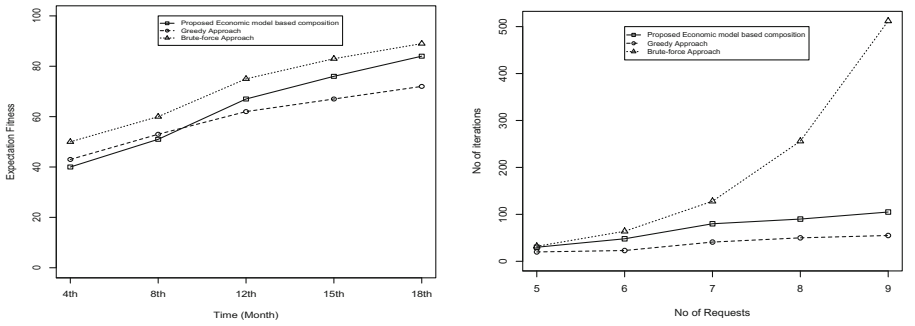
generate the long-term behavior between operation cost and utilization using the dynamic electricity price in Table 1. The SLA violation fee is 20 % of the revenue credited to consumers. The resource constraints are set by allowing maximum 100 units for each attributes. We use the resource allocation algorithm in [11] as the utility factor in Eq. 6.

4.2 Efficiency of the Economic Model Based Composition

At first, we evaluate the performance of the propose DBN in the economic model. We model the 18 months (Jan 2012–Sep 2013) electricity price with the proposed DBN. The predicted price from the DBN is compared with the actual price using Normalized Root Mean Square Error (NRMSE) defined in Eq. 11.

$$NRMSE = \sqrt{\frac{\sum_{i=1}^{n}(\frac{PR(q)-AC(q)}{AC(q)_{max}-AC(q)_{min}})^2}{n}} \tag{11}$$

Figure 1(b) depicts the closeness (0.3 NRMSE) of the prediction toward actual behavior. Figure 2(a) depicts the efficiency of the long-term composition over the greedy approach. We observe that the greedy approach only maximizes the economic fitness at the beginning, while the proposed economic model based service composition maximizes the fitness in a steady rate over the period of composition (Fig. 2(a)). Although brute-force produces the best composition in Fig. 2(b), it is not applicable as its time complexity is (2^N). The proposed approach is not polynomial with respect to request size (Fig. 2(a)). Although it takes



(a) (b)

Fig. 2. (a) The long-term composition fitness (b) Effect of request size in time complexity

more computation time than the greedy approach, the difference is negligible in real world implementation.

5 Conclusion

In summary, we propose a novel dynamic service composition framework for IaaS providers to gain their long-term economic expectations. The proposed long-term economic model evaluates the long-term economic behavior of service compositions. Experimental results show that our approach is more profitable than the greedy approach and suitable for runtime implementation.

Acknowledgements. This research was made possible by NPRP 7-481-1-088 grant from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A.: Above the clouds: A berkeley view of cloud computing. Tech, UC Berkeley (2009)
2. Ye, Z., Bouguettaya, A., Zhou, X.: QoS-aware cloud service composition based on economic models. In: Liu, C., Ludwig, H., Toumani, F., Yu, Q. (eds.) Service Oriented Computing. LNCS, vol. 7636, pp. 111–126. Springer, Heidelberg (2012)
3. Ye, Z., Mistry, S., Bouguettaya, A., Dong, H.: Long-term qos-aware cloud service composition using multivariate time series analysis. IEEE Trans. Serv. Comput. **1**(99), 1–16 (2014)
4. Mistry, S., Bouguettaya, A., Dong, H., Qin, A.K.: Predicting dynamic requests behavior in long-term iaas service composition. In: Proceedings of ICWS (2015)
5. Wu, L., Garg, S., Buyya, R.: Sla-based resource allocation for software as a service provider (saas) in cloud environments. In: Proceedings of CCGrid, pp. 195–204 (2011)

6. Ergu, D., Kou, G., et al.: The analytic hierarchy process: task scheduling and resource allocation in cloud. J. Supercomput. **64**, 835–848 (2013)
7. Dash, D., Kantere, V., Ailamaki, A.: An economic model for self-tuned cloud caching. In: Proceedings of ICDE, pp. 1687–1693 (2009)
8. Thanakornworakij, T., Nassar, R. et al.: An economic model for maximizing profit of a cloud service provider. In: Proceedings of ARES, pp. 274–279 (2012)
9. Goiri, Í., Guitart, J., Torres, J.: Economic model of a cloud provider operating in a federated cloud. Inf. Syst. Front. **14**(4), 827–843 (2012)
10. Murphy, K.P.: "Dynamic bayesian networks: representation, inference and learning," Ph.D. dissertation, University of California, Berkeley (2002)
11. Wu, L., Garg, S., Buyya, R.: Sla-based admission control for a software-as-a-service provider in cloud computing environments. J. Comput. Syst. Sci. **78**(5), 1280–1299 (2012)
12. Ye, Z., Zhou, X., Bouguettaya, A.: Genetic algorithm based QoS-aware service compositions in cloud computing. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) DASFAA 2011, Part II. LNCS, vol. 6588, pp. 321–334. Springer, Heidelberg (2011)
13. Qian, H.: PivotalR: a package for machine learning. R J. **6**(1), 57 (2014)
14. Reiss, C., Wilkes, J., Hellerstein, J.L.: Google cluster-usage traces: format + schema. Google Inc., Mountain View, CA, USA, Technical report (2011)
15. Jiang, W., Lee, D., Hu, S.: Large-scale longitudinal analysis of soap-based and restful web services. In: Proceedings of ICWS, pp. 218–225 (2012)
16. ESAA, "Electricity prices in australia," Aus B. of Statistics, Technical report 02 (2000)