

## Research Article

# A QoS-Satisfied Prediction Model for Cloud-Service Composition Based on a Hidden Markov Model

**Qingtao Wu, Mingchuan Zhang, Ruijuan Zheng, Ying Lou, and Wangyang Wei**

*College of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China*

Correspondence should be addressed to Qingtao Wu; [wuqingtao.cn@hotmail.com](mailto:wuqingtao.cn@hotmail.com)

Received 31 May 2013; Accepted 4 July 2013

Academic Editor: Yuxin Mao

Copyright © 2013 Qingtao Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Various significant issues in cloud computing, such as service provision, service matching, and service assessment, have attracted researchers' attention recently. Quality of service (QoS) plays an increasingly important role in the provision of cloud-based services, by aiming for the seamless and dynamic integration of cloud-service components. In this paper, we focus on QoS-satisfied predictions about the composition of cloud-service components and present a QoS-satisfied prediction model based on a hidden Markov model. In providing a cloud-based service for a user, if the user's QoS cannot be satisfied by a single cloud-service component, component composition should be considered, where its QoS-satisfied capability needs to be proactively predicted to be able to guarantee the user's QoS. We discuss the proposed model in detail and prove some aspects of the model. Simulation results show that our model can achieve high prediction accuracies.

## 1. Introduction

Cloud computing is a term used to refer to the use of widespread, shared computing resources. It is an alternative to having local servers handle computing applications. Cloud computing groups together a large number of computing servers and other resources, often offering their combined capacity on an on-demand, pay-per-cycle basis. The end users of a cloud computing network usually have no idea where the servers are physically located—they just open their applications and start working [1–3].

In general, the service resources in cloud computing will include hardware resources (e.g., processors, storage, and networking) and software resources (e.g., web servers, databases, message queuing systems, and monitoring systems). Cloud service types can be abstracted into three layers, namely software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) [4]. Hardware and software resources form the basis for delivering IaaS and PaaS. The SaaS layer at the top focuses on application services by making use of services provided by the lower layers. PaaS/SaaS services are often developed and provided by third-party services providers who are different from the IaaS provider.

Therefore, matching the cloud-service components to the users' quality of service (QoS) is very important because user

experience is a principal reason for promoting the development of cloud computing. For each user request, the provider should select an appropriate composition of cloud-service components to serve the user if there is no single cloud-service component that satisfies the user's QoS perfectly. This will require predictions to be made about the QoS satisfaction for compositions of cloud-service components.

Wang et al. [5] propose a composition method for selecting cloud-based web services from candidate services, changing a single service into a more powerful composite service, which uses the Skyline operator and Particle Swarm Optimization. Javadi et al. [6] investigate cloud-computing resource provision to extend the computing capacity of local clusters in the presence of failures. Their three steps in resource provision include resource brokering, dispatch sequences, and scheduling. Benouaret et al. [7] present an approach to composing Data Web services automatically while taking into account user preferences, which is based on fuzzy sets and top-k optimization. Liu et al. [8] introduce a scheme to address the particularities of manufacturing-resource service composition and optimization, where the user's QoS is considered. Zhang et al. [4] present an investigation of an intelligent decision-support system for selecting cloud-based infrastructure services to choose the best mix

of service offerings from an abundance of possibilities. Although those papers discuss the composition of cloud-service components, QoS-satisfied predictions are not considered.

Huang et al. [9] introduce a method for addressing the problem of composing a sequence of service components for QoS-guaranteed service provision in a virtualization-based cloud-computing environment. Huang et al. [10] designs a suboptimal resource-allocation system in a cloud-computing environment, and a corresponding prediction mechanism is realized by using support-vector regression to estimate the resource utilization. Di and Wang [11] propose a fully distributed, VM-multiplexing resource-allocation scheme to manage decentralized resources. Jiang et al. [12] propose a new method for cloud-capacity planning, with the goals of utilizing the physical resources fully and providing an integrated system with intelligent cloud-capacity prediction.

The Markov model (MM), particularly the hidden Markov model (HMM), has been shown to be a good technique for solving prediction problems. Zhang and Pathirana [13] present an adaptive HMM for identifying underlying path losses. Choi et al. [14] present a sparsely correlated HMM for assessing multiple genomic datasets. Botev et al. [15] present a versatile Monte Carlo method for estimating multi-dimensional integrals, with application to rare-event probability estimation. Xie et al. [16] present a new structurally discrete approach to predicting network traffic called the nested hidden semi-Markov model, which includes a nested latent semi-Markov chain and one observable discrete stochastic process. However, further research into QoS-satisfied capability should be considered.

This paper focuses on QoS-satisfied predictions for compositions of cloud-service components, based on the HMM and our previous work [17–19]. The remainder of this paper is organized as follows. In Section 2, the basic model used in later discussion is presented. In Section 3, we propose a QoS-satisfied prediction model and explain it in detail. In Section 4, the simulation and analysis are discussed. Finally, we conclude the paper in Section 5.

## 2. Basic Model

The MM is a statistical model with a wide area of application. Many other Markov models are derived fundamentally from MM, including the HMM and the semi-Markov model. The HMM is used extensively for performance modeling and performance-prediction analysis, where the HMM can predict the future state of a target system based on its current state. In reality, because the relationship between the observed time and the observed state is not one to one, a group of probability distributions for two stochastic processes are involved, called the HMM.

In an HMM, the states are not observable, but when we visit a state, an observation is recorded that is a probabilistic function of the state. We assume a discrete observation in each state from the set

$$\{v_1, v_2, \dots, v_M\} : b_j(m) \triangleq \Pr(O_t = v_m | q_t = S_j), \quad (1)$$

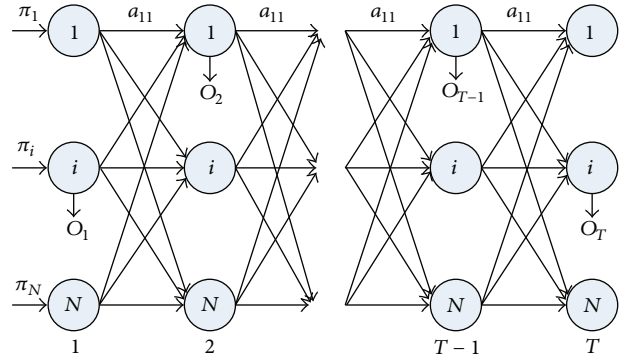


FIGURE 1: An example of an observable model for an HMM.

where  $b_j(m)$  is the *observation* or *emission probability*. We also assume a homogeneous model for which the probabilities do not depend on  $t$ . The values thus observed constitute an observation sequence  $O$ . The state sequence  $Q$  is not observed directly (being “hidden”), but it should be possible to infer it from the observation sequence  $O$ . An example of an observable model for an HMM is shown in Figure 1, where  $M = N$  and  $b_j(m) = 1$  if  $j = m$  and  $b_j(m) = 0$  otherwise. To summarize and formalize, an HMM has the following elements:

- (1)  $N$ : the number of states in the model

$$S = \{S_1, S_2, \dots, S_N\} \quad (2)$$

- (2)  $M$ : the number of distinct observation symbols in the *alphabet*

$$V = \{v_1, v_2, \dots, v_M\}. \quad (3)$$

- (3) State transition probabilities:

$$A = [a_{ij}], \quad \text{where } a_{ij} = \Pr(q_{t+1} = S_j | q_t = S_i). \quad (4)$$

- (4) Observation probabilities:

$$B = [b_j(m)], \quad \text{where } b_j(m) = \Pr(O_t = v_m | q_t = S_j). \quad (5)$$

- (5) Initial state probabilities:

$$\Pi = [\pi_i], \quad \text{where } \pi_i = \Pr(q_1 = S_i). \quad (6)$$

$N$  and  $M$  are implicitly defined by the other parameters, leaving  $\lambda = (A, B, \Pi)$  as the parameter set for an HMM. Given  $\lambda$ , the model can be used to generate an arbitrary number of observation sequences of arbitrary length, but we are usually interested in the other direction, namely, that of estimating the parameters of the model given a training set of sequences.

## 3. Proposed QoS-Satisfied Prediction Model for Cloud-Service Composition

In this section, we propose a prediction model for cloud-service composition based on an HMM. We first present an overview of cloud-service composition and then give details of the prediction model.

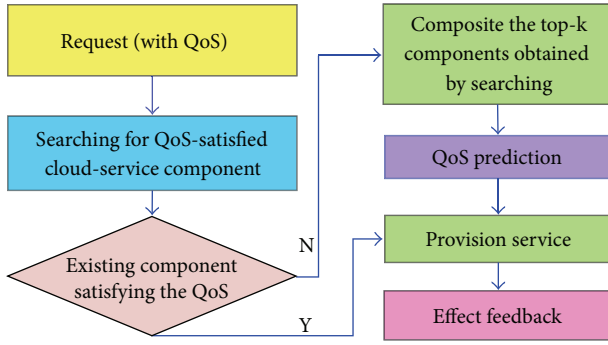


FIGURE 2: General flowchart for cloud-service provision.

### 3.1. An Overview of Cloud-Service Composition and Matching.

The key roles played in a cloud environment are those of the service user and the service provider. The cloud-service user needs anytime, anywhere QoS-satisfied and low-cost services that are flexible and easy to use. The important hurdles to users adopting cloud services involve security, availability, and reliability. We should therefore assess the QoS-satisfied capability for each service provision generated by matching cloud-service components with the users' QoS. Sometimes, if a single cloud-service component does not satisfy the users' QoS, multiple cloud-service components should be composed to provide a more complex service. General flowcharts for cloud services and for a cloud-service component-matching model are shown in Figures 2 and 3, respectively.

**3.2. QoS-Satisfied Prediction Model.** We propose a QoS-satisfied prediction model based on an HMM to predict whether a composition of cloud-service components can satisfy the user's QoS. Using the basic HMM model, we assume that the QoS capability of cloud-service components is a state set  $\{v_1, v_2, \dots, v_M\} : b_j(m) \triangleq \Pr(O_t = v_m \mid q_t = S_j)$ , where  $b_j(m)$  is the observation probability of obtaining  $v_m$  ( $m = 1, 2, \dots, M$ ) when the composition state is  $S_j$  and  $O$  is a sequence of obtained  $v_m$  values. To reduce the complexity of calculation for  $\Pr(O \mid \lambda)$ , we define a forward variable

$$\alpha_t(i) \triangleq \Pr(O_1, O_2, \dots, O_t, q_t = S_i \mid \lambda), \quad 1 \leq i \leq N. \quad (7)$$

The three stages of its recursion are shown in Figure 4(a) and described as follows.

(1) Initialization:

$$\begin{aligned} \alpha_1(i) &\triangleq \Pr(O_1, q_1 = S_i \mid \lambda) \\ &= \Pr(O_1 \mid q_1 = S_i, \lambda) \Pr(q_1 = S_i \mid \lambda) \\ &= \pi_i b_i(O_1). \end{aligned} \quad (8)$$

(2) Recursion:

$$\begin{aligned} \alpha_{t+1}(j) &\triangleq \Pr(O_1, \dots, O_{t+1}, q_{t+1} = S_j \mid \lambda) \\ &= \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}). \end{aligned} \quad (9)$$

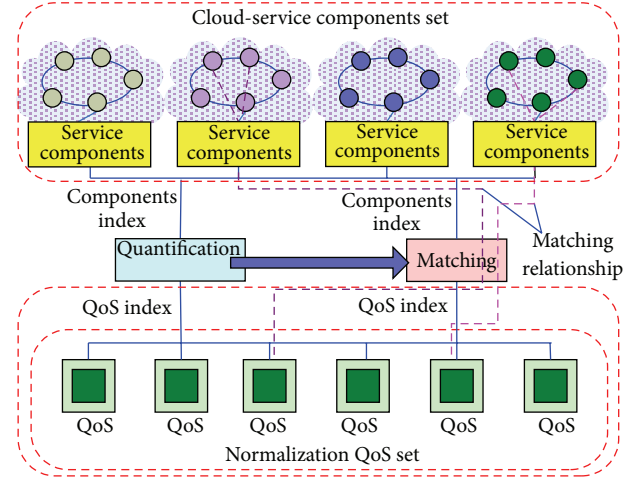


FIGURE 3: Cloud-service component-matching model.

*Proof*

$$\begin{aligned} \alpha_{t+1}(j) &\triangleq \Pr(O_1, \dots, O_{t+1}, q_{t+1} = S_j \mid \lambda) \\ &= \Pr(O_1, \dots, O_{t+1} \mid q_{t+1} = S_j, \lambda) \Pr(q_{t+1} = S_j \mid \lambda) \\ &= \Pr(O_1, \dots, O_t \mid q_{t+1} = S_j, \lambda) \\ &\quad \cdot \Pr(O_{t+1} \mid q_{t+1} = S_j, \lambda) \Pr(q_{t+1} = S_j \mid \lambda) \\ &= \Pr(O_1, \dots, O_t, q_{t+1} = S_j \mid \lambda) \\ &\quad \cdot \Pr(O_{t+1} \mid q_{t+1} = S_j, \lambda) \\ &= \Pr(O_{t+1} \mid q_{t+1} = S_j, \lambda) \\ &\quad \cdot \sum_{i=1}^N \Pr(O_1, \dots, O_t, q_t = S_i, q_{t+1} = S_j \mid \lambda) \\ &= \Pr(O_{t+1} \mid q_{t+1} = S_j, \lambda) \\ &\quad \cdot \sum_{i=1}^N \Pr(O_1, \dots, O_t, q_{t+1} = S_j \mid q_t = S_i, \lambda) \\ &\quad \cdot \Pr(q_t = S_i \mid \lambda) \\ &= \Pr(O_{t+1} \mid q_{t+1} = S_j, \lambda) \\ &\quad \cdot \sum_{i=1}^N \Pr(O_1, \dots, O_t, q_t = S_i \mid \lambda) \\ &\quad \cdot \Pr(q_{t+1} = S_j \mid q_t = S_i, \lambda) \\ &= \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}). \end{aligned} \quad (10)$$

□

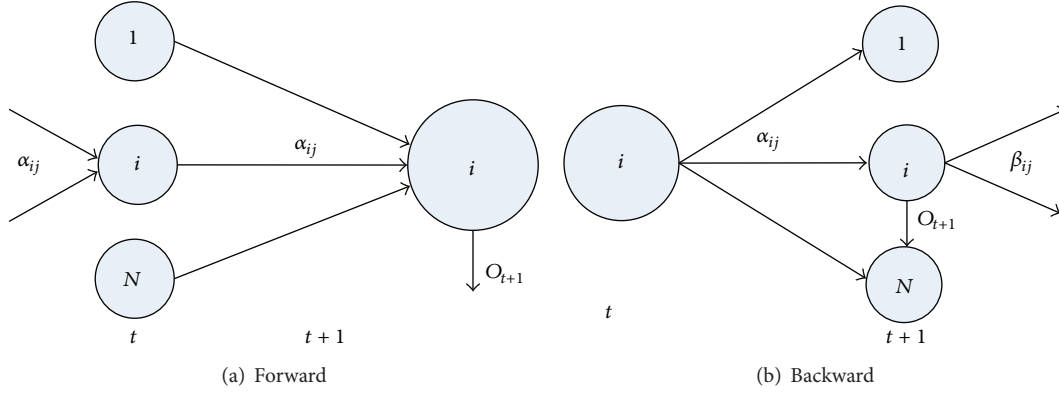


FIGURE 4: Forward-backward procedure: (a) computation of  $\alpha_t(j)$ ; (b) computation of  $\beta_t(j)$ .

(3) End:

$$\Pr(O | \lambda) = \sum_{i=1}^N \Pr(O, q_T = S_i | \lambda) = \sum_{i=1}^N \alpha_T(i). \quad (11)$$

Similarly, we define a backward variable

$$\beta_t(i) \triangleq \Pr(O_{t+1}, \dots, O_T | q_t = S_i, \lambda). \quad (12)$$

The three stages of its recursion are shown in Figure 4(b) and described as follows.

(1) Initialization:

$$\beta_T(i) = 1. \quad (13)$$

(2) Recursion:

$$\begin{aligned} \beta_t(i) &\triangleq \Pr(O_{t+1}, \dots, O_T | q_t = S_i, \lambda) \\ &= \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j). \end{aligned} \quad (14)$$

*Proof*

$$\begin{aligned} \beta_t(i) &\triangleq \Pr(O_{t+1}, \dots, O_T | q_t = S_i, \lambda) \\ &= \sum_{j=1}^N \Pr(O_{t+1}, \dots, O_T, q_{t+1} = S_j | q_t = S_i, \lambda) \\ &= \sum_{j=1}^N \Pr(O_{t+1}, \dots, O_T | q_{t+1} = S_j, q_t = S_i, \lambda) \\ &\quad \cdot \Pr(q_{t+1} = S_j | q_t = S_i, \lambda) \end{aligned}$$

$$\begin{aligned} &= \sum_{j=1}^N \Pr(O_{t+1} | q_{t+1} = S_j, q_t = S_i, \lambda) \\ &\quad \cdot \Pr(O_{t+2}, \dots, O_T | q_{t+1} = S_j, q_t = S_i, \lambda) \\ &\quad \cdot \Pr(q_{t+1} = S_j | q_t = S_i, \lambda) \\ &= \sum_{j=1}^N \Pr(O_{t+1} | q_{t+1} = S_j, \lambda) \\ &\quad \cdot \Pr(O_{t+2}, \dots, O_T | q_{t+1} = S_j, \lambda) \\ &\quad \cdot \Pr(q_{t+1} = S_j | q_t = S_i, \lambda) \\ &= \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j). \end{aligned} \quad (15)$$

□

(3) End:

$$\Pr(O | \lambda) = \sum_{i=1}^N \beta_1(i). \quad (16)$$

From these forward and backward variables, we have

$$\begin{aligned} \Pr(O | \lambda) &= \sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \\ &1 \leq t \leq T-1. \end{aligned} \quad (17)$$

We therefore obtain  $\lambda^* = \arg \max_{\lambda} \Pr(O | \lambda)$ , which is a functional extremum problem. Because the length of the training sequence is limited, there is no optimal method for estimating  $\lambda$ . Therefore, recursion is used to achieve a maximum local value for  $\Pr(O | \lambda)$ , thereby obtaining the model parameter set  $\lambda = (A, B, \Pi)$ .

We define  $\xi_t(i, j)$  as the probability of being in  $S_i$  at time  $t$  and in  $S_j$  at time  $t+1$ , given the whole observation sequence  $O$  and the model parameter set  $\lambda$ . Then

$$\begin{aligned} \xi_t(i, j) &\triangleq \Pr(q_t = S_i, q_{t+1} = S_j | O, \lambda) \\ &= \frac{\alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j) a_{ij}}{\sum_{k=1}^N \sum_{l=1}^N \alpha_t(k) a_{kl} b_l(O_{t+1}) \beta_{t+1}(l)}. \end{aligned} \quad (18)$$

*Proof*

$$\begin{aligned} \xi_t(i, j) &\triangleq \Pr(q_t = S_i, q_{t+1} = S_j | O, \lambda) \\ &= \Pr(O | q_t = S_i, q_{t+1} = S_j, \lambda) \\ &\quad \cdot \frac{\Pr(q_t = S_i, q_{t+1} = S_j | \lambda)}{\Pr(O | \lambda)} \\ &= \Pr(O | q_t = S_i, q_{t+1} = S_j, \lambda) \\ &\quad \cdot \Pr(q_{t+1} = S_j | q_t = S_i, \lambda) \\ &\quad \cdot \frac{\Pr(q_t | \lambda)}{\Pr(O | \lambda)} \\ &= \Pr(O_1, \dots, O_t | q_t = S_i, \lambda) \\ &\quad \cdot \Pr(O_{t+1} | q_{t+1} = S_j, \lambda) \\ &\quad \cdot \Pr(O_{t+2}, \dots, O_T | q_{t+1} = S_j, \lambda) \\ &\quad \cdot \frac{a_{ij} \Pr(q_t = S_i | \lambda)}{\Pr(O | \lambda)} \\ &= \Pr(O_1, \dots, O_t, q_t = S_i | \lambda) \Pr(O_{t+1} | q_{t+1} = S_j, \lambda) \\ &\quad \cdot \frac{\Pr(O_{t+2}, \dots, O_T | q_{t+1} = S_j, \lambda) a_{ij}}{\Pr(O | \lambda)} \\ &= \frac{\alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j) a_{ij}}{\Pr(O | \lambda)} \\ &= \frac{\alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j) a_{ij}}{\sum_{k=1}^N \sum_{l=1}^N \Pr(q_t = S_k, q_{t+1} = S_l, O | \lambda)} \\ &= \frac{\alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j) a_{ij}}{\sum_{k=1}^N \sum_{l=1}^N \alpha_t(k) a_{kl} b_l(O_{t+1}) \beta_{t+1}(l)}. \end{aligned} \quad (19)$$

Thus, the probability of being in  $s_i$  at time  $t$  is therefore given by

$$\gamma_t(i) = \Pr(O, q_t = S_i | \lambda) = \sum_{j=1}^N \xi_t(i, j), \quad (20)$$

where  $\sum_{t=1}^{T-1} \xi_t(i, j)$  denotes the mathematical expectation of transition from state  $S_i$ . Therefore, the probability of transition from  $S_i$  to  $S_j$  is given by

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \quad (21)$$

In state  $S_j$ , the probability of observing  $v_m$  is given by

$$\hat{b}_j(m) = \frac{\sum_{t=1}^T \gamma_t(j) 1_{(O_t=v_m)}}{\sum_{t=1}^T \gamma_t(j)}. \quad (22)$$

We therefore obtain a new model  $\hat{\lambda} = (\hat{A}, \hat{B}, \hat{\Pi})$ , where  $\hat{A} = [\hat{a}_{ij}]$ ,  $\hat{B} = [b_j(m)]$ ,  $\hat{\Pi} = [\hat{\pi}_i]$ , and  $\hat{\pi}_i = \gamma_1(i)$ . We can prove  $\Pr(O | \hat{\lambda}) \geq \Pr(O | \lambda)$ . The training process is repeated, and the model parameters are adjusted gradually until  $\Pr(O | \hat{\lambda})$  converges. Finally, we obtain the prediction model  $\hat{\lambda}$ , which is used to assess the QoS-satisfied capability for compositions of cloud-service components.  $\square$

#### 4. Stimulation Experiment and Analysis

To check the feasibility of the proposed QoS-satisfied prediction model, we constructed a simulation experiment based on a cloud-computing experimental platform at our university. The simulation system included more than 100 cloud-service components, such as adaptable components and servers, and the open-source Eucalyptus system was adopted.

Using our simulation system, we designed three cloud storage services whose QoS were different to check the prediction model. First, we trained the prediction model by adjusting the model parameters gradually, using a machine learning algorithm based on a support vector machine (SVM). In the SVM, the penalty factor  $c$ , the kernel function parameter  $g$ , and the boundary range  $p$  were set to 1024, 1024, and 0.0097, respectively. After obtaining the model parameter set  $\lambda$ , we achieved the prediction model  $\hat{\lambda}$ . We then made predictions for the three cloud storage services by using the prediction model and acquiring their simulated results. The simulation experiment was run 10 times to produce average results. The QoS-satisfied capabilities for the prediction values and observation values of the three cloud storage services ( $A$ ,  $B$ , and  $C$ ) are shown in Figures 5 and 6, respectively. The comparison between actual and observed values for service  $B$  is shown in Table 1, where the selected observation times from 74 onwards show the acquisition of a stable observed effect.

Note that the error is small (no more than 0.01%). This is because there is a stable network environment, without any other factors' effects in our simulated system. In reality, a larger error would be expected. However, it should be able to satisfy the user's QoS requirements in most cases.

#### 5. Conclusions

Cloud computing refers to the use of shared computing resources and is an alternative to having local servers handle



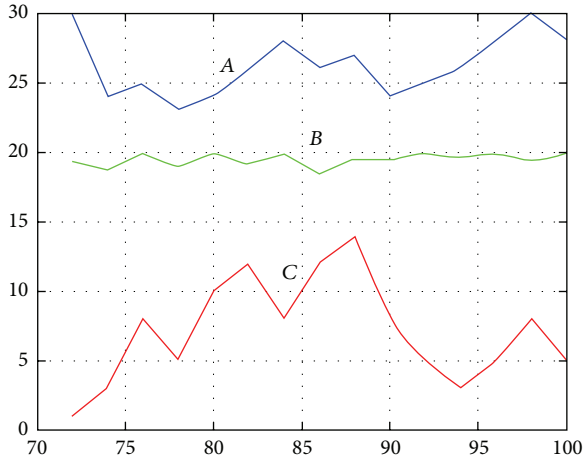


FIGURE 5: Prediction values.

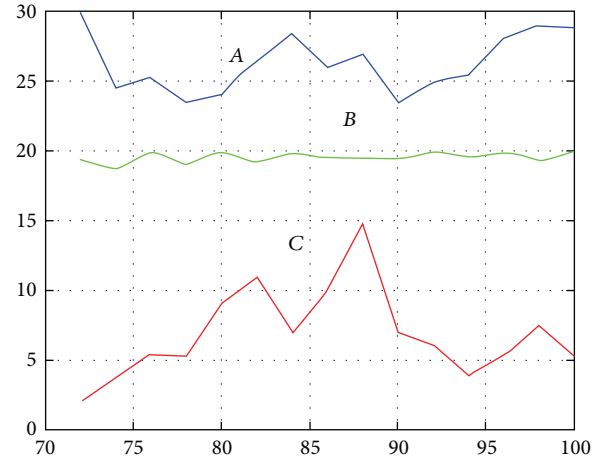


FIGURE 6: Observation values.

TABLE 1: Comparison of prediction results.

Time	Actual value	Observation value	Error (%)
74	19.4	19.3995	0.003
76	18.7	18.7011	0.006
78	20	19.9994	0.003
80	19	19.001	0.005
82	20	19.9993	0.004
84	19.2	19.2008	0.004
86	19.9	19.8991	0.005
88	18.5	18.5011	0.006
90	19.5	19.4991	0.005
92	19.4	19.3986	0.007
94	20	19.999	0.005
96	19.5	19.4993	0.004
98	20	19.9991	0.005
100	19.3	19.2987	0.007

user applications. A match between cloud-service components and users' QoS is therefore very important if user experience is the basis for promoting the development of cloud computing. If no single cloud service satisfies the user's QoS perfectly, a composition of multiple cloud-service components should be considered, which should also include predicting the QoS satisfaction of the composite service. This paper presents a QoS-satisfied prediction model based on an HMM to assess the QoS-satisfied capability for compositions of components. We discuss the proposed model in detail and prove aspects of the model. Simulation results show that our model can achieve high prediction accuracies. In future work, we will introduce heterogeneous cloud services into the system and develop the prediction model to make it a better fit with real applications.

## Acknowledgments

The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have

helped improve the quality of this paper. This work was supported in part by National Natural Science Foundation of China (NSFC) under Grant no. 61003035 and no. U1204614, in part by the Plan for Scientific Innovation Talent of Henan Province under Grant no. 124100510006, and in part by the Science and Technology Development Programs of Henan Province under Grant no. 112102210187, in part by the Youth Foundation of Henan University of Science and Technology under Grant no. 2011QN51.

## References

- [1] D. Nurmi, R. Wolski, C. Grzegorzczak et al., "The eucalyptus open-source cloud-computing system," in *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID '09)*, pp. 124–131, Shanghai, China, May 2009.
- [2] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [3] L. Wang, R. Ranjan, J. Chen, and B. Benatallah, *Cloud Computing: Methodology, Systems, and Applications*, CRC Press, 2011.
- [4] M. Zhang, R. Ranjan, A. Haller, D. Georgakopoulos, and P. Strazdins, "Investigating decision support techniques for automating Cloud service selection," in *Proceedings of the 4th International Conference on Cloud Computing Technology and Science*, pp. 759–764, Taipei, China, December 2012.
- [5] S. Wang, Q. Sun, H. Zou, and F. Yang, "Particle swarm optimization with skyline operator for fast cloud-based web service composition," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 116–121, 2013.
- [6] B. Javadi, P. Thulasiraman, and R. Buyya, "Enhancing performance of failure-prone clusters by adaptive provisioning of cloud resources," *The Journal of Supercomputing*, vol. 63, no. 2, pp. 467–489, 2013.
- [7] K. Benouaret, D. Benslimane, A. Hadjali, and M. Barhamgi, "Top-k web service compositions using fuzzy dominance relationship," in *Proceedings of the IEEE International Conference on Services Computing (SCC '11)*, pp. 144–151, Washington, England, July 2011.
- [8] W.-N. Liu, B. Liu, and D.-H. Sun, "A conceptual framework for dynamic manufacturing resource service composition and

- optimization in service-oriented networked manufacturing,” in *Proceedings of the International Conference on Cloud and Service Computing (CSC '11)*, pp. 118–125, Hong Kong, China, December 2011.
- [9] J. Huang, Y. Liu, R. Yu, Q. Duan, and Y. Tanaka, “Modeling and algorithms for QoS-aware service composition in virtualization-based cloud computing,” *IEICE TRANSACTIONS on Communications*, vol. 96, no. 1, pp. 10–19, 2013.
- [10] C. J. Huang, C. T. Guan, H. M. Chen et al., “An adaptive resource management scheme in cloud computing,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 382–389, 2013.
- [11] S. Di and C. Wang, “Dynamically optimizing multi-attribute execution efficiency on self-organizing cloud,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 3, pp. 464–478, 2013.
- [12] Y. Jiang, C. Perng, T. Li, and R. Chang, “Self-adaptive cloud capacity planning,” in *Proceedings of the IEEE Ninth International Conference on Services Computing*, pp. 73–80, Honolulu, HI, USA, June 2012.
- [13] H. Zhang and P. Pathirana, “Uplink power control via adaptive hidden Markov model based pathloss estimation,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 4, pp. 657–665, 2013.
- [14] H. Choi, D. Fermin, A. I. Nesvizhskii, D. Ghosh, and Z. S. Qin, “Sparsely correlated hidden Markov models with application to genome-wide location studies,” *Bioinformatics*, vol. 29, no. 5, pp. 533–541, 2013.
- [15] Z. I. Botev, P. L’Ecuyer, and B. Tuffin, “Markov chain importance sampling with applications to rare event probability estimation,” *Statistics and Computing*, vol. 23, no. 2, pp. 271–285, 2013.
- [16] Y. Xie, J. Hu, S. Tang, and X. Huang, “A forward-backward algorithm for nested hidden semi-Markov model and application to network traffic,” *The Computer Journal*, vol. 56, no. 2, pp. 229–238, 2013.
- [17] R. Zheng, M. Zhang, Q. Wu, S. Sun, and J. Pu, “Analysis and application of bio-inspired multi-net security model,” *International Journal of Information Security*, vol. 9, no. 1, pp. 1–17, 2010.
- [18] R. Zheng, Q. Wu, M. Zhang, G. Li, J. Pu, and H. Wang, “A self-optimization mechanism of system service performance based on autonomic computing,” *Computer Research and Development*, vol. 48, no. 9, pp. 1676–1684, 2011.
- [19] M.-C. Zhang, Q.-T. Wu, R.-J. Zheng, W.-Y. Wei, and G.-F. Li, “Research on grade optimization self-tuning method for system dependability based on autonomic computing,” *Journal of Computers*, vol. 7, no. 2, pp. 333–340, 2012.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

