

A QoS-satisfied Prediction Model for Cloud-service Composition Based on Hidden Markov Model

<http://dx.doi.org/10.3991/ijoe.v9i3.2678>

Qingtao Wu, Mingchuan Zhang, Ruijuan Zheng and Wangyang Wei
Henan University of Science and Technology, Luoyang, P. R. China.

Abstract—There are various significant issues in cloud computing, such as service provision, service matching and service assessment, which have attracted researchers' attentions recently. QoS play an increasingly important role during the procedure of cloud-based service provision for seamless and dynamic integration of cloud-service components. In this paper, we focus on the QoS-satisfied prediction for cloud-service composited components and present a QoS-satisfied prediction model based on hidden Markov model. For a general process of cloud-service provision, if the user's QoS could not be satisfied only by one cloud-service component, the component composition should be considered to provide to user, where the QoS-satisfied capability of composited components need to be proactively predicted to guarantee the user's QoS. We discuss the proposed model in detail and proof the model partly. The simulation results show that our model can obtain rather high prediction accuracy rate.

Index Terms—Cloud-service, Composition, Hidden Markov Model, QoS-satisfied

I. INTRODUCTION

Cloud computing is a way of referring to the use of shared computing resources, and it is an alternative to having local servers handle applications. Cloud computing groups together large numbers of computing servers and other resources and typically offers 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 spin up their application and start working [1-3].

Generally, the service resources of cloud computing include: hardware resources (e.g. CPU, storage and network) and software resources (e.g. web server, databases, message queuing systems and monitoring systems). Cloud service types can be abstracted into three layers: 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 top layer focuses on application services (SaaS) by making use of services provided by the lower layers. PaaS/SaaS services are often developed and provided by third party service providers who are different from the IaaS providers.

Therefore, the match between cloud-service components and users' QoS is very important since user experience is the base to promote the development of cloud computing. For a user's request, provider should

select the proper cloud-service component(s) to serve the user. If there is not a cloud-service component which satisfy the user's QoS perfectly. Composition of multiple cloud-service components should be considered as well as the QoS-satisfied prediction of composited cloud-service components.

Wang *et al.* [5] propose a composited method to select cloud-based web services from amount of candidates for composing single service into a more powerful composite service based on 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 and three steps in the resource provision including resource brokering, dispatch sequences and scheduling are considered. Benouaret *et al.* [7] present an approach to automatically compose Data Web services while taking into account user preferences 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. Zhang *et al.* [4] present investigating on an intelligent decision support system for selecting cloud-based infrastructure services to select the best mix of service offering from an abundance of possibilities. Although those papers discuss the composition of cloud-service components, the QoS-satisfied prediction is not considered.

Huang *et al.* [10] introduce a method to address the problem on how to compose a sequence of service components for QoS guaranteed service provisioning in a virtualization-based Cloud computing environment. Paper [11] designs a sub-optimal resource allocation system in a cloud computing environment and a corresponding prediction mechanism is realized by using support vector regressions to estimate the number of resource utilization. Di *et al.* [12] propose a fully distributed, VM-multiplexing resource allocation scheme to manage decentralized resources. Jiang *et al.* [13] propose a new method for cloud capacity planning with the goal of fully utilizing the physical resources as well as an integrated system with intelligent cloud capacity prediction. Besides, Markov model (MM), especially hidden Markov model (HMM), has been shown to be a good technique for solving the prediction problem. Zhang *et al.* [14] present an adaptive HMM to identify the underlying pathloss. Choi *et al.* [15] present sparsely correlated HMM to assess multiple genomic datasets. Botev *et al.* [16] present a versatile Monte Carlo method for estimating multi-dimensional integrals, with applications to rare-event probability

estimation. Xie *et al.* [17] present a new structural discrete approach named nested hidden semi-Markov model which includes a nested latent semi-Markov chain and one observable discrete stochastic process to prediction network traffic. However, further researches for QoS-satisfied capability should be considered.

This paper focuses on the QoS-satisfied prediction for cloud-service composited components based on HMM and our previous work [18-20]. The rest of this paper is organized as follows: in section II, the basic model used in later discussion is presented; in section III, we propose QoS-satisfied prediction model and explicate it in detail; in section IV, the simulation and analysis are discussed; finally, we conclude this paper in section V.

II. BASIC MODEL

MM is a statistical model with extensive application. Many other Markov models are derived fundamentally from MM, e.g. HMM and semi-Markov model, where HMM are extensively used for performance modeling and performance prediction analysis. HMM could predict the future state of target system based on its current state. In reality, since the relationship between observed time and observed state is not one to one, it relates to a group of probability distribution with two stochastic processes, which is called HMM.

In 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

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

where $b_j(m)$ is the *observation* or *emission probability*.

We again assume a homogeneous model in which the probabilities do not depend on t . The values thus observed constitute the observation sequence O . The state sequence Q is not observed, that is what makes the model "hidden", but it should be inferred from the observation sequence O . An example of observable model for HMM is shown in Fig. 1, where $M=N$ and $b_j(m)$ is 1 if $j=m$ and 0 otherwise. To summarize and formalize, HMM has the following elements:

- 1) N : Number of state in the model

$$S=\{S_1, S_2, \dots, S_N\}$$

- 2) M : Number of distinct observation symbols in the *alphabet*

$$V=\{v_1, v_2, \dots, v_M\}$$

- 3) State transition probabilities:

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

- 4) Observation probabilities:

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

- 5) Initial state probabilities:

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

N and M are implicitly defined in the other parameters so $\lambda=(A, B, \Pi)$ is taken as the parameter set of an HMM. Given λ , the model can be used to generate an arbitrary number of observation sequences of arbitrary length, but as usual, we are interested in the other

direction, that of estimating the parameters of the model given a training set of sequences.

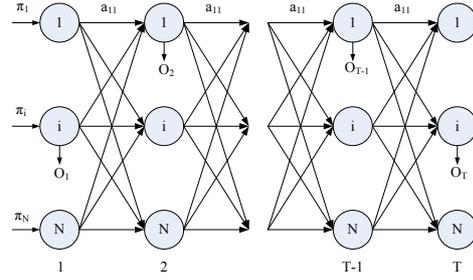


Figure 1 An example of observable model for HMM

III. PROPOSED QoS-SATISFIED PREDICTION MODEL FOR CLOUD-SERVICE COMPOSITION

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

A. Overview of Cloud-service Composition and Matching

The key roles in a cloud environment include 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 adoption of cloud-service for users lie in: security, availability and reliability. Therefore, we should assess the QoS-satisfied capability for each provision service which is generated according to the matching between cloud-service component and users' QoS. Sometimes, if only one cloud-service component may not satisfy the users' QoS, multiple cloud-service components should be composited to provide more complicated service. The general flow of cloud-service and cloud-service components matching model are shown in Fig. 2 and Fig. 3, respectively.

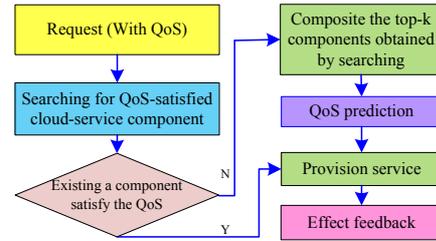


Figure 2 General flow of cloud-service provision

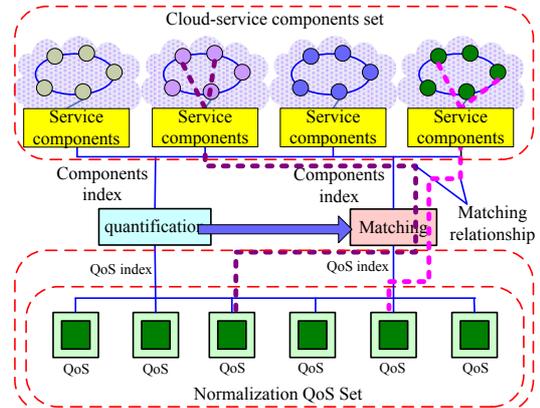


Figure 3 Cloud-service components matching model

B. QoS-satisfied Prediction Model

We propose a QoS-satisfied prediction based on HMM to predict whether the composited cloud-service components can satisfy the user's QoS. Based on the basic HMM model, we assume that 1) the QoS capability of cloud-service components is a state set $\{v_1, v_2, \dots, v_M\}$: $b_j(m) @Pr(O_t = v_m | q_t = S_j)$, where $b_j(m)$ is the observation probability obtaining $v_m(m=1,2,\dots,M)$ when composition state is S_j , O is a sequence of obtained v_m . In order to reduce the complexity of calculating $Pr(O|\lambda)$, we define forward variable as

$$\alpha_i(i) @Pr(O_1, O_2, \dots, O_i, q_i = S_i | \lambda), 1 \leq i \leq N$$

Its recursion is shown in Fig. 4(a) and as follow.

1) Initialization:

$$\begin{aligned} \alpha_1(i) @Pr(O_1, q_1 = S_i | \lambda) \\ = Pr(O_1 | q_1 = S_i, \lambda) Pr(q_1 = S_i | \lambda) \\ = \pi_i b_i(O_1) \end{aligned}$$

2) Recursion:

$$\begin{aligned} \alpha_{t+1}(j) @Pr(O_1, \dots, O_t, q_{t+1} = S_j | \lambda) \\ = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \end{aligned}$$

Proof:

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

3) End:

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

Similarly, we define backward variable as

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

And its recursion is shown in Fig. 4(b) and as follows

1) Initialization:

$$\beta_T(i) = 1$$

2) Recursion:

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

Proof:

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

3) End:

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

From forward variable and backward variable, we have

$$Pr(O|\lambda) = \sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j),$$

$$1 \leq t \leq T-1$$

Thus, we obtain $\lambda^* = \arg \max_{\lambda} Pr(O|\lambda)$, which is a functional extremum problem. Since the length of training sequence is limited, there is no the optimal method to estimate λ . Therefore, recursion is used to achieve the maximum value of $Pr(O|\lambda)$ in locality and obtain model parameter $\lambda=(A,B,\Pi)$.

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 model parameter λ

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

Proof:

$$\xi_t(i,j) @Pr(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$

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

thus, the probability of being in S_i at time t is shown as

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

where, $\sum_{t=1}^{T-1} \xi_t(i, j)$ denotes the mathematical expectation of transfer number form state S_i . Therefore, the probability of transiting from S_i to S_j is shown as

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

In state S_j , the probability of observing v_m is shown as

as

$$\hat{b}_j(m) = \sum_{t=1}^T \gamma_t(j) \mathbf{1}_{(O_t=v_m)} / \sum_{t=1}^T \gamma_t(j)$$

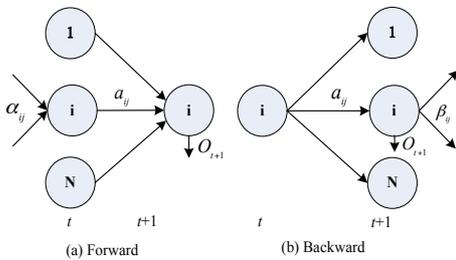


Figure 4 Forward-backward procedure. (a) computation of $\alpha_t(j)$; (b) computation of $\beta_t(j)$.

Thus, we obtain a new model $\hat{\lambda} = (\hat{A}, \hat{B}, \hat{\Pi})$, where $\hat{A} = [\hat{a}_{ij}]$, $\hat{B} = [b_j(m)]$, $\hat{\Pi} = [\hat{\pi}_i]$, $\hat{\pi}_i = \gamma_1(i)$. We can

proof $\Pr(O|\hat{\lambda}) \geq \Pr(O|\lambda)$. The training process is repeated and the model parameter is adjusted gradually until $\Pr(O|\hat{\lambda})$ gets convergence. Finally, we gain the prediction model $\hat{\lambda}$, which is used to assess QoS-satisfied capability for composited cloud-service components.

IV. SIMULATION EXPERIMENT AND ANALYSIS

In order to validate the feasibility of proposed QoS-satisfied Prediction Model, we construct a simulation experiment system based on cloud computing experiment platform in our university. The simulation system includes more than 100 cloud-service components, adapting components, servers, etc., where the open source Eucalyptus is adopted.

We composite three cloud storage services whose QoS are different using our simulation system to verify the prediction model. Firstly, we train the prediction model by adjusting the model parameter gradually, where machine learning algorithm of Support Vector Machine (SVM) is adopted. In SVM, the penalty factor c , kernel function parameter g and boundary range p are set to 1024, 1024 and 0.0097, respectively. After obtaining the model parameter λ , we achieve the prediction model $\hat{\lambda}$. Then, we predict the three cloud storage services A , B and C using the prediction model and acquire their simulation results, where simulation experiment is run 10 times to calculate their average values. The QoS-satisfied capability of both prediction value and observation value of three cloud storage services are shown in Fig. 5 and Fig. 6, respectively, and the comparison of service B is shown in Table I, where we select observation time from 74 to acquire a stable observed effect.

We can note that the error is rather small (less than 0.01%), since there is a stable network environment and without other factors' effects in simulation system. If it is in reality, the error will increase a little. However, it can satisfy the user's QoS in most cases.

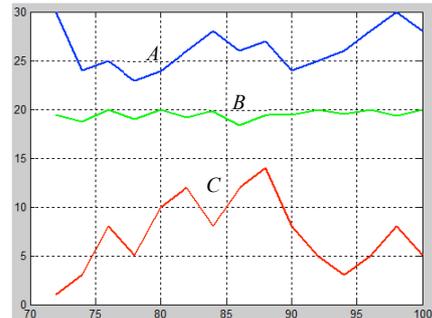


Figure 5 Prediction value

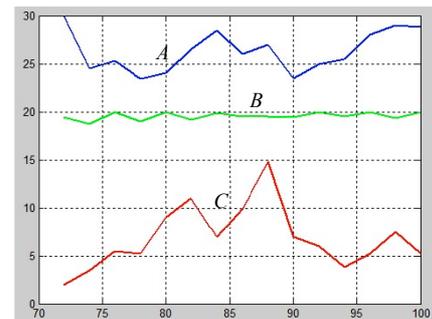


Figure 6 Observation value

TABLE I.
PREDICTION RESULTS COMPARISON

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

V. CONCLUSIONS

Cloud computing is a way of referring to the use of shared computing resources, and it is an alternative to having local servers handle applications. Therefore, the match between cloud-service components and users' QoS is very important since user experience is the base to promote the development of cloud computing. If there is not a cloud-service component which satisfy the user's QoS perfectly. Composition of multiple cloud-service components should be considered as well as the QoS-satisfied prediction of composited cloud-service components. This paper present a QoS-satisfied prediction model based on HMM to assess the QoS-satisfied capability for composited components. We discuss the proposed model in detail and proof the model partly. The simulation results show that our model can obtain rather high accuracy rate in prediction. In future work, we will introduce the heterogeneous cloud-service into the system as well as advancing the prediction precision to make it fit with reality well.

REFERENCES

- [1] D. Nurmi, R. Wolski, and C. Grzegorzcyk. 2009, "The Eucalyptus Open-source Cloud Computing System," Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, Shanghai, China, 2009.
- [2] M. Armbrust, A. Fox, R. Griffith, etc., "A view of Cloud Computing," Communications of the ACM Magazine, Vol. 53, No. 4, pp. 50-58, 2010. <http://dx.doi.org/10.1145/1721654.1721672>
- [3] L. Wang, R. Ranjan, J. Chen, etc., "Cloud Computing: Methodology, Systems, and Applications," Edited Book, 844 Pages, CRC Press, Taylor and Francis Group, Oct. 2011.
- [4] M. Zhang, R. Ranjan, A. Haller A, etc., "Investigating decision support techniques for automating Cloud service selection," IEEE 4th International Conference on Cloud Computing Technology and Science, Taipei, China, pp. 759-764, Dec. 2012.
- [5] S. Wang, Q. Sun, H. Zou, etc., "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. <http://dx.doi.org/10.1007/s11036-012-0373-3>
- [6] B. Javadi, P. Thulasiraman, 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. <http://dx.doi.org/10.1007/s11227-012-0826-2>
- [7] K. Benouaret, D. Benslimane, A. Hadjali, etc., "Top-k web service compositions using fuzzy dominance relationship," IEEE International Conference on Services Computing, Washington, England, pp. 144-151, July. 2011.
- [8] W. Liu, B. Liu and D. Sun, "A conceptual framework for dynamic manufacturing resource service composition and optimization in service-oriented networked manufacturing," International Conference on Cloud and Service Computing (CSC), Hong Kong, China, pp. 118-125, Dec. 2011.
- [9] J. Huang, Y. Liu, R. Yu, etc., "Modeling and Algorithms for QoS-Aware Service Composition in Virtualization-Based Cloud Computing," IEICE TRANSACTIONS on Communications, vol. 96, no. 1, pp. 10-19, Jan. 2013. <http://dx.doi.org/10.1587/transcom.E96.B.10>
- [10] C. Huang, C. Guan, H. Chen, etc., "An adaptive resource management scheme in cloud computing," Engineering Applications of Artificial Intelligence, vol. 26, no. 1, pp. 382-389, Jan. 2013. <http://dx.doi.org/10.1016/j.engappai.2012.10.004>
- [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, Mar. 2013. <http://dx.doi.org/10.1109/TPDS.2012.144>
- [12] Y. Jiang, C. Perng, T. Li, etc., "Self-adaptive Cloud Capacity Planning," IEEE the International Conference on Services Computing, Honolulu, USA, pp. 73-80, Jun. 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, Apr. 2013. <http://dx.doi.org/10.1109/TMC.2012.39>
- [14] H. Choi, D. Fermin, A. Nesvizhskii, etc., "Sparsely correlated hidden Markov models with application to genome-wide location studies," Bioinformatics, vol. 29, no. 5, pp. 533-541, Jan. 2013. <http://dx.doi.org/10.1093/bioinformatics/btt012>
- [15] Z. 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, Mar. 2013. <http://dx.doi.org/10.1007/s11222-011-9308-2>
- [16] Y. Xie, J. Hu, S. Tang, etc., "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, Feb. 2013. <http://dx.doi.org/10.1093/comjnl/bxs124>
- [17] R. Zheng, M. Zhang, Q. Wu, etc., "Analysis and application of bio-inspired multi-net security model," International Journal of Information Security, vol.9, no.1, pp.1-17, Jan. 2010. <http://dx.doi.org/10.1007/s10207-009-0091-4>
- [18] R. Zheng, Q. Wu, M. Zhang, etc., "A self-optimization mechanism of system service performance based on autonomic computing," Journal of Computer Research and Development, vol. 48, no. 9, pp.1676-1684, Sept. 2011.
- [19] M. Zhang, Q. Wu, R. Zheng, etc., "Research on Grade Optimization Self-Tuning Method for System Dependability Based on Autonomic Computing," Journal of Computers, vol.7, no.2, pp.333-340, Feb. 2012. <http://dx.doi.org/10.4304/jcp.7.2.333-340>

AUTHORS

Q.T. Wu is with the Electronic & Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China (e-mail: wqt8921@126.com).

M.C. Zhang, R.J. Zheng and W.Y. Wei are with the Electronic & Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China.

This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. U1204614 and No. 61003035, and 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, and in part by the Youth Foundation of Henan University of Science and Technology under Grant No. 2011QN51. Received 6 January 2013. Published as resubmitted by the authors 12 June 2013.