An EC Services System Using Evolutionary Algorithm

Whe Dar Lin

The Overseas Chinese Institute of Technology
Dept of Information Management,
No. 100, Chiao Kwang Road, Taichung 40721, Taiwan

Abstract. Our new evolutionary method allows electronic commerce (EC) services on distinct distribution channels. Launching EC services on the Internet require careful on mobile agents. It supports EC transition flows written in XML. Our algorithm resolves the concurrent data-accessing problem among EC services databases. To create a better algorithm, we have analyzed a variety of transaction schemes compatible with standards and developed a modeling framework on which maintaining good consistency. With our EC transaction method, we can make use of different techniques and organize an EC framework with clients, agents, and EC application servers all included form an integrated EC system management mechanism. Our proposed system can improve the relationship between EC service systems and transaction agents for supply-chain management.

1 Introduction

To keep mobile agents on track toward making a purchase, EC systems must provide an effective function of local applications with applications running on remote servers. M-services do pose challenges to database management and transactions on EC services platforms in order to support greater workgroups and achieve better organizational productivity. The ultimate goal is to provide a richer and more user-friendly environment of information by integrating the user's desktop facilities with information exchange and collaboration infrastructures including groupware platforms and shared database servers. In a business setting, these information services are typically part of an EC service system [1], [2], [6], [8], [9].

One of the advantages of mobile agents can control their own shared resources. A commercial deal usually involves several transactions including the transfer of contract documents, billing, and settlement of payment. Sometimes several transactions need to be integrated, as when billing and settlement are to be processed at the same time. In addition, the definition of priority transactions is required for defining the entire commercial deal, so that should any individual transaction fail, the entire deal can be discarded [4], [5].

Evolutionary methods have been applied to a variety of different Web-based problems. In this paper, an algorithm for EC services system based on an evolutionary model is proposed.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3036, pp. 549–552, 2004.

2 Our Evolutionary Algorithm

In our evolutionary model, the reinforcements can be either positive or negative, depending on whether the realized channel cost is greater or less than what the EC services need. Given the evolutionary approach method set E_i of EC agent i, where $E_i = \{e_{i,1}, e_{i,2}, e_{i,3}, \cdots, e_{i,Mi}\}$ respectively, there are alternative pure evolutionary approach to be performed by EC agent i, (i = 1, …, M). EC agent i at each period uses an evolutionary approach method, and the state of the system in period t is denoted by $C_{t,i}$. Note that here in this place $C_{t,i} =(C_{t,i}(e_{i,1}), C_{t,i}(e_{i,2}), C_{t,i}(e_{i,3}), \ldots, C_{t,i}(e_{i,Mi}))$ is the probability distribution of the evolutionary approach method set E_i in period t by EC agent i. If EC agent i plays evolutionary method $C_{t,i}$ in period t, then the resultant loading balance value is $C_{loading}(t, i, e_{t,i})$. The EC agent's communication channel cost is denoted by $C_{linking}(t, i, e_{t,i})$, and we set the loading balance value as $C_{balance}(t, i, e_{t,i}) =C_{loading}(t, i, e_{t,i}) - C_{linking}(t, i, e_{t,i})$. The C_{index} value is iif($e_{i,k} =e_{i,t}$, 1, 0)=$C_{index.}$ Then, for i = 1,…,N and k =1, 2,…,Mi, the system state evolves in the following way:

$$C_{t+1,i}(e_{k,i}) = (1+\left|C_{balance}(t,i,e_{t,i})\right|) * C_{t,i}(e_{k,i}) + C_{index} * C_{balance}(t,i,e_{t,i}) \tag{1}$$

Thus, it can be seen that if $C_{balance}(t, i, e_{t,i})$ is positive, that means the EC agent is pleased with the outcome, and then the probability associated with the strategy will increase. In our proposed algorithm, EC transactions can be calculated in terms of link capacity, buffer size, queue length, etc. In addition, we can even update the switching function on the arrival of every transaction. The key idea behind our proposed algorithm is to update the switching probability according to the loading strategy rather than the instantaneous or average loading weight, maintaining a single probability $C_{probablity}(t, i, e_{t,i})$ to transfer enqueued transactions.

I: Computing switching function

$$\begin{aligned} &iif\,(L_{load}(t,i,e_{t,i}) \ge L_{threshhold}, 1, (L-(L_{threshhold} - L_{load}(t,i,e_{t,i})))/L) \\ &\quad = L_{switch}(t,i,e_{t,i}) \end{aligned} \tag{2}$$

The system loading value in period t is denoted by $C_{load}(t, i, e_{t,i})$. We set a loading weight threshold, $C_{threshhold}$.

II: Computing moving probability

$$\begin{aligned} &iif\,(L_{switch}(t,i,e_{t,i}) \ge L_{switch}(t-1,i,e_{t-1,i}), \\ &L_{switch}(t,i,e_{t,i}) + (1-C_{probablity}(t,i,e_{t,i})) * C_{balance}(t,i,e_{t,i}), \\ &L_{switch}(t,i,e_{t,i}) + C_{probablity}(t,i,e_{t,i}) * C_{balance}(t,i,e_{t,i})) \\ &\quad = L_{switch}(t+1,i,e_{t+1,i}) \end{aligned} \tag{3}$$

This result can be derived from equations listed in Sect. 3. Thus, when the outcome satisfies the EC transaction services, the loading probability is increased. However, the switching probability is increased when the EC services are dissatisfied.

In the next section, we shall present our simulation results on our proposed algorithm and see how it compares with other algorithms in the same network environment. We will show the validity and features of our proposed EC services algorithm.

3 Performance with Our Evolutionary Algorithm

Simulation results show that our system outperforms such existing EC services schedulers as earliest deadline, highest value and hierarchical earliest deadline when an application requires an EC transaction model.

We examined EC transactions under various conditions. According to the metrics of Commit times and throughput, our method has the best performance for distributed EC Web services using EC transaction models.

The setting for these basic parameters is based on our experiment, we varied the arrival rate from 1 transactions/second to 5 trans/sec.

Table 1. Commit time simulation results of different method

Channel availability	Commit time (millisec)				
	0.4	0.6	0.8	1	1.2
Earliest deadline	123	99	92	89	89
Highest value	115	90	91	87	86
Hierarchical earliest deadline	111	86	80	69	66
Our Evolutionary Algorithm	100	81	72	60	54

Table 2. Throughput simulation results of different method

Channel availability	Throughput (transaction/sec)				
	0.4	0.6	0.8	1	1.2
Earliest deadline	0.10	0.15	0.15	0.20	0.20
Highest value	0.15	0.15	0.15	0.20	0.20
Hierarchical earliest deadline	0.15	0.20	0.20	0.20	0.25
Our Evolutionary Algorithm	0.20	0.25	0.30	0.35	0.40

Tables 1 and 2 show the commit time and throughput results for real time EC services transactions. The results for EC services transactions. The performance orders are Our Evolutionary Algorithm > Hierarchical earliest deadline > Highest value> Earliest deadline.

Simulation results show that our system outperforms the others on throughput and commit time. An EC transaction based on evolutionary algorithm, to each transaction resides in the ready queue with the highest will be executed. The appropriate setting for the communication delay of the real time transactions can meet their loading balance value on time under the simulation results.

To begin with, the consideration of the loading characteristic in Web services gives a higher weight in the formula in the evolutionary model at the arrival of a transaction, since such a transaction requires an expensive cost for accessing data objects in the database. However, the loading policy also depends on the reward ratio and loading balance value as well as the slack time of the system. In addition, the communication delays in our evolutionary algorithm will result in a slightly higher weight for a remote transaction; hence, a local transaction will have a better chance to be executed completely under the adjustment of a transaction's reward ratio.

4 Conclusion

In this paper, we have presented a new algorithm to handle electronic commerce (EC) transactions on Web-based systems. Evolutionary methods have been used to solve a wide variety of Web-based systems problems. We have demonstrated that it is capable of offering smooth transaction services at an extremely low loss rate with little delay in supply chain management. We can enable the EC Web server to adapt to various network conditions and traffic characteristics intelligently. Simulation results show that our system outperforms others on throughput and commit time. It prevents the queue from turning into overflow and decreases the loss rate due to buffer overflow. All the parameters used in the algorithm can be derived and adjusted by using measured and estimated information. Indeed, the complexity of our new algorithm is lower than those of many other algorithms. Our evolutionary method responses rapidly to the changes of the network load by adjusting the switching probability quickly. The concepts presented in this paper can be further developed into a set of networks that will help identify the best design alternative for high balance loading management based on the characteristics and parameters of given transactions on EC service applications in supply chain management. The performance of our method in complex network topologies is not yet clear. We will work on that in the future. In addition, we shall also focus on the development of new service algorithms and differentiated service support in supply chain management.

References

1. C. M. Weng and P. W. Huang, “More Efficient Location Tracking in PCS Systems Using a Novel Distributed Database System,” IEEE transactions on vehicular technology, Vol. 51, No.4, pp277-289, 2002.
2. D. Fudenberg and D.K. Levine, The Theory of Learning in Games, The MIT Press, 1998.
3. El-Sayed, A.A., Hassanein, H.S., and El-Sharkawi, M.E. “Effect of shaping characteristics on the performance of transactions.” Information and Software Technology 43(10): 579-590,2001.
4. Haritsa, J.R., Ramamritham, K., and Gupta, R. “The PROMPT real-time commit protocol.” IEEE Trans. Parallel and Distributed Systems 11(2):160-181, 2000.
5. Jain, R. “The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.” WILEY, 1991.
6. J. W. Weibull, Evolutionary Game Theory, The MIT Press, 1995.
7. K. K. Leung, Y. Levy, “Global Mobility Management by Replicated Databases in Personal Communication Networks,” IEEE Journal on selected areas in communications, Vol. 15, No. 8, pp1582-1596, 1997.
8. R. Somegawa, K. Cho, Y. Sekiya, and S. Yamaguchi, “The Effect of Server Placement and Server Selection for Internet Services,” IEICE Trans. on Communications, Vol.E86-B, No.2, PP.542-552, 2003.
9. V. Kanitkar and A. Delis, “Real-Time Processing in Client-Server Databases,” IEEE transactions on computers, Vol. 51, No.3, pp269-288, 2002.