An Increase Web Services Performance Method

Whe Dar Lin

The Overseas Chinese Institute of Technology, Dept of Information Management,
No. 100, Chiao Kwang Road, Taichung 40721, Taiwan
darlin@ocit.edu.tw

Abstract. In this paper, our loading balance processes electronic commerce transactions to increase Web services performance. Our method includes a loading balance value to resolve the concurrent data problem among EC services databases Web systems. In the new algorithm proposed here, a loading evaluation is introduced and utilized to make the efficiency of the result. The simulation results show that our system outperforms existing schedulers such as branch and bound algorithm, and the shifting bottleneck heuristic when an application requires an EC transaction model. We have analyzed a variety of transaction schemes compatible with other kinds of protocol standards and developed a modeling framework on which we can selectively extract the merits of these schemes while maintaining good consistency. Simulation results show that our system outperforms existing schedulers such as throughput and commit time when an application requires an EC transaction model. The electronic commerce transactions hype has barely subsided, the media, venture capitalists, and stock markets have already moved on to the mobile Internet. Knowing a change is coming is loading balance, predicting its shape and form for electronic commerce transactions to increase Web services performance on PC-centric models to mobile and person-centric techniques.

1 Introduction

The structure of computing is changing. EC Web systems rely increasingly on the coupling of local applications with applications running on remote Web servers. Assisted with low-power, low-cost, and portable computing Web systems such as laptops and personal digital assistants (PDAs), real-time agents can now work anywhere at any time. Mobility and portability do pose challenges to database management and EC transactions on EC Web systems, for they require the use of real-time information services designed to provide coherent and reliable information access to wide varieties of data originating from separate information sources in order to support greater workgroups and achieve better organizational productivity. In the EC era, there is a general trend towards the partnership in between EC Web systems so as to ensure better EC transaction application efficacy and efficiency [1, 3, 7, 11, 16].

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 Web systems and shared database servers. In a business setting, these information services are typically part of a real-time database management system [2, 4, 6, 8, 11, 12].

The multi-channel customer revolution coupled with the development of mobile technology in its to have a profound effect. A commercial deal usually involves sev-

D. Lowe and M. Gaedke (Eds.): ICWE 2005, LNCS 3579, pp. 130–135, 2005.

eral 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 [13, 14, 15, 16].

Where the uncertainty lies is in precisely when and how loading will be imposed and which will emerge as EC transactions. Evolutionary methods have been applied to a variety of different problems. In this paper, an algorithm for EC transaction management based on an evolutionary model is proposed. This novel algorithm generates a loading balance value for the purpose of improving the responsiveness to EC transactions on the Internet Web servers while maintaining a reasonable degree of queueing performance in case of a buffer overflow. Our simulation results have demonstrated the effectiveness of the proposed algorithm and proved the superiority over other algorithms. As a wish, the nature of interaction and speed is bound to change of transaction [5, 9, 10].

2 Our Model

The adoption of mobile technological innovation by the transaction world is seldem a neat, linear process. 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 A_i of EC agent i, where $A_i = \{a_{i,1}, a_{i,2}, a_{i,3}, \cdot\cdot\cdot, a_{i,Mi}\}$ 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 $S_{t,i}$.

Note that here in this place $S_{t,i} =(S_{t,i}(a_{i,1}), S_{t,i}(a_{i,2}), S_{t,i}(a_{i,3}), \ldots, S_{t,i}(a_{i,Mi}))$ is the probability distribution of the evolutionary approach method set A_i in period t by EC agent i. If EC agent i plays evolutionary method $S_{t,i}$ in period t, then the resultant loading balance value is $S_{LV}(t, i, a_{t,i}, S_D)$. The EC agent's communication channel cost is denoted by $S_{MV}(t, i, a_{t,i}, S_D)$, and we set the loading balance value as $S_{BV}(t, i, a_{t,i}, S_D) =S_{LV}(t, i, a_{t,i}, S_D) - S_{MV}(t, i, a_{t,i}, S_D)$. The S_D value is $\max(a_{i,k}, a_{i,t})$. Then, for i = 1,…,N and k =1, 2,…,Mi, the system state evolves in the following way:

$$S_{t+1,i}(a_{k,i}) = S_D * S_{BV}(t,i,a_{t,i},S_D) + (1 + |S_B(t,i,a_{t,i},S_D)|) * S_{t,i}(a_{k,i}) \qquad (1)$$

Thus, it can be seen that if $S_{BV}(t, i, a_{t,i}, S_D)$ 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 $S_{PV}(t, i, a_{t,i}, S_D)$ to transfer enqueued transactions.

I: Computing Switching Function

$$S_{SV}(t,i,e_{t,i},S_D) = (S_{BV} - (S_{TV} - S_{LV}(t,i,a_{t,i},S_D)))/S_{BV} \qquad (2)$$

The system loading value in period t is denoted by $S_{LV}(t, i, e_{t,i}, S_D)$. We set a loading weight threshold, S_{TV}.

II: Computing Moving Probability

$$S_{SV}(t+1, i, a_{t+1,i}, S_D) = S_{PV}(t, i, a_{t,i}, S_D) * S_{BV}(t, i, a_{t,i}, S_D) + S_{SV}(t, i, a_{t,i}, S_D) \tag{3}$$

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.

III: Our Loading Balance Method

1: Offspring generation.
2: EC system loading increase.
3: New parent selection among loading balance value.
4: Loading balance value updating.
5: Satisfied output the parents and loading balance value.

3 Our Loading Balance Method Performance

Certain media on Internet portray the transition to the mobile services. It prevents other transactions from accessing a shared data object before transaction is done. Simulation results show that our system outperforms such existing schedulers as branch and bound algorithm and the shifting bottleneck heuristic when an application requires an EC transaction model.

(1) Branch and bound algorithm: Branch and bound algorithm is enumerative search procedures based on construction of a tree of partial solutions. The tree contains entire set of feasible solutions in the leaves, while nodes inside the tree represent partial solution.
(2) Shifting bottleneck heuristic: The shifting bottleneck heuristic takes advantage of that scheduling problem with release time can be solved by bottleneck identification.
(3) Loading Balance Method: Our new scheduler is an evolutionary model to dynamically implement available transaction strategies by employing a loading balance value.

The offered value refers to the value given by the application when a transaction is submitted, and the final-obtained value indicates the net value after the transaction is successfully completed.

We examined real-time EC transactions under various conditions. According to the metrics of Commit times and throughput, the loading balance method has the best performance for distributed EC Web services using EC transaction models. In this experiment, we varied the arrival rate from 1 transactions/second to 5 trans/sec. Channel availability is number of bandwidth for EC transactions.

Table 1. Commit time simulation results of different method

Scheduler	Commit time (millisec)				
(Channel availability)	0.4	0.6	0.8	1	1.2
Branch and bound algorithm	195	175	130	100	90
Shifting bottleneck heuristic	215	200	170	135	105
Loading balance method	165	140	115	75	50

Table 2. Throughput simulation results of different method

Scheduler	Throughput (transaction/sec)				
(Channel availability)	0.4	0.6	0.8	1	1.2
Branch and bound algorithm	0.10	0.10	0.15	0.15	0.15
Shifting bottleneck heuristic	0.10	0.10	0.10	0.10	0.15
Loading balance method	0.20	0.25	0.30	0.30	0.35

Tables 1 and 2 show commit time and throughput results for EC services transactions. The performance orders are Loading balance method > Branch and bound algorithm> Shifting bottleneck heuristic, respectively.

The reason is that branch and bound algorithm and shifting bottleneck heuristic are both designed for a centralized case, and the impact of communication delays is not considered a scheduling factor. Such a phenomenon will frequently occur, as the traffic load gets higher. Branch and bound algorithm and shifting bottleneck heuristic perform poorly in commit time because they do not consider transaction relations. The performance of loading balance method is good at a high load because its scheduling policy in the under an overload situation.

Simulation results show that our system outperforms the others on throughput, and commit time. An EC transaction based on loading balance method, 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.

The reasons for the superiority of loading balance method over the other algorithms are as follows. 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 loading balance method 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. Furthermore, loading balance method also takes care of the relations between transactions, taking into account the concept of slack time distribution.

4 Conclusion

Now, it is a portal to cyberspace an entry point to a worldwide network of information exchange and business transactions. We have presented a loading balance method to handle EC transactions on EC services systems for penetrate most areas of our daily life. Our loading balance method processes electronic commerce transactions to increase Web services performance only very limited support for processing Web content. Our loading balance method adjusts the transaction blocking probability according to the value of the switching function and its loading balance value.

Using loading balance method can enable the EC Web server to adapt to various network conditions and traffic characteristics intelligently and integration of data, information, knowledge, process and applications within business. Our results show that our system outperforms others on throughput and commit time, it beats such schedulers as branch and bound algorithm and the shifting bottleneck heuristic when an application requires an EC transaction model as enterprise resource planning from different product of implementing one link per supplier can be linked to a marketplaces.

Our loading balance method prevents the queue from turning into overflow and decreases the loss rate due to buffer overflow for reasons of cost and quality to adopt product as customer relationship management. Our loading balance method responses rapidly to the changes of the network load by adjusting the switching probability quickly within existing corporate information technology infrastructures solution. 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 real time transactions on EC service applications as merge often require large-scale integration of existing information technology infrastructures. We shall focus on the development of new Web-base service management methods with QoS and differentiated service support for mobile services more openness, flexibility and dynamics.

References

1. Abbott, R. and Garcia-Molina, H. “Scheduling EC transactions.” ACM SIGMOD Record 17(1):71-81, 1988.
2. B.Krishnam urthy and J.Rexford, Web protocols and practice: HTTP/1.1, networking protocols, caching, and traffic measurement, 1st ed., Addison Wesley, 2001.
3. D. Friedman, “On economic applications of evolutionary game theory,” J. Evolutionary Economics, vol.8, no.1, pp.15–43, 1998.
4. 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.
5. E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: Empirical results,” Evolutionary Computation, vol.8, no.2, pp.173–195, 2000.
6. 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.
7. Huang, J. and Stankovic, J. “Real-Time Buffer Management.” COINS TR 90-65. 1990.
8. H.Zh u and T.Y ang, “Cachuma: Class-based cache management for dynamic web content,” Proc. IEEE INFOCOM 2001, pp.1215–1224, 2001.
9. J. Knowles and D.W. Corne, “Approximating the nondominated front using the Pareto archived evolution strategy,” Evolutionary Computation, vol.8, no.2, pp.149–172, 2000.
10. J. W. Weibull, Evolutionary Game Theory, The MIT Press, 1995.
11. 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.

12. M. Arlitt and T. Jin, “Workload characterization of the 1998 world cup web site,” IEEE Network, vol.14, no.3, pp.30–37, 2000.
13. M. Karam and F. Tobagi, “Rate and queue control random drop (RQRD): A buffer management scheme for Internet routers,” Proc. Globecom’00, vol.1, no.1, pp.316–322, San Francisco, CA, Nov. 2000.
14. R. Pan, B. Prabhakar, and K. Psounis, “CHOKe: A stateless active queue management scheme for approximating fair bandwidth allocation,” Proc. IEEE Infocom’00, vol.2, no.1, pp.942–951, March 2000.
15. U. Bodin, O. Schelen, and S. Pink, “Load-tolerant differentiation with active queue management,” SIGCOMM Computer Communication Review, vol.30, no.3, pp.4–16, July 2000.
16. Whe Dar Lin, “EC Transactions Use Different Web-based Web systems,” Lecture Notes in Computer Science, LNCS-2658, pp1059-1068, 2003.