An Intelligent Trust-based Information Retrieval Mechanism in P2P Networks

Jun-qing Li
College of Computer Science, Liaocheng University, Liaocheng, 252059, People's Republic of China
lijunqing@lcu.edu.cn

Quan-ke Pan
College of Computer Science, Liaocheng University, Liaocheng, 252059, People's Republic of China
panquanke@lcu.edu.cn

Sheng-xian Xie
College of Computer Science, Liaocheng University, Liaocheng, 252059, People's Republic of China
xsx@lcu.edu.cn

Yu-ting Wang
College of Computer Science, Liaocheng University, Liaocheng, 252059, People's Republic of China
wangyuting@lcu.edu.cn

ABSTRACT

In this paper, we give a novel trust-based P2P information retrieval system, which use ant colony optimization (ACO) algorithm to compute the trust value of a peer and adjust the trust value with time. The new mechanism works as follows: firstly, an initialized trust value was assigned to each peer in the system; secondly, when a client peer send a request to its neighbors, it will create an ant named F_{ant}, and the F_{ant} selects a right route based on the selection probability in the route table; thirdly, when a F_{ant} reaches the server peer, a new ant named B_{ant} will be created; at last, the B_{ant} will come back on the same way and adjust the trust value of the peers on the route based on a certain rule. To verify the efficiency of our novel mechanism, we also give a detailed experiment. The experiment results conclude that the new trust-based information retrieval platform works well and robust.

Key Words: Peer-to-peer; Information retrieval; Ant colony optimization

1. Introduction

Peer-to-peer (P2P) networks have been developed and used in many scopes. With P2P systems, the following scenes can be truth: with P2P chat software such as ICQ, a person can talk with any one without facing with him; with P2P download software such as Thunder, a person can download any file from another person even they are stranger; with P2P stream media software such as PPLive, a person can see a film without considering who provide the service. There are many surprising things in P2P networks, which we can not find in the traditional networks such as B/S (Browser/Server) and C/S (Client/Server). All peers in P2P networks can act as a server when they provide services for other peers; they can also get services from other peers, at that time, we call these peers clients. In the P2P networks, every peer can join or leave the network platform without any constraint. However, many researches pointed that P2P networks are faced with lots of security problems such as file pollution, malicious peers, freeriders, and Sybil attackers. If we select a malicious peer for some files, we will get some pollution files with a high probability, and if we select a freerider, we will get nothing. There are millions of peers in the P2P networks, how to select a nice peer as the service provider is the key issue in the information retrieval mechanism.

The first and the simplest mechanism for the information retrieval is Flooding, which is used in Gnutella system. The main idea behind the Flooding is to send the request information to all neighbors around the peer without considering the difference between the peers. The Flooding is used for many years because of its simplicity. However, with the development of the P2P system, more and more malicious peers occur in the P2P network. The result of the Flooding mechanism can be concluded as follows: more and more network bandwidth were used for propagated the request information even a great number of the receiver does not have the answer; there is not any difference between the malicious peer and the good peer, which make the malicious even bad and the good peer less enthusiasm to provide service.

The second mechanism for the information retrieval is the Top-*k*, which selects top *k* neighbors to forward the request messages. The Top-*k* neighbors can be selected by several rules such as the relationship between the request information and the content of the neighbor peer, the trust value of the neighbor peer and the bandwidth or the computational capability of the neighbor peer. Because we select the Top-*k* neighbors instead of the entire neighbor peers, the network bandwidth consumed by the information retrieval can be saved.

The last and efficient mechanism is the trust-based P2P platform. In this platform, every peer was assigned a certain trust value. A client peer selects a candidate server peer based on the trust value of the server. The traditional trust-based P2P platform has many disadvantages including the lack of flexible, consumption of the network bandwidth and lack of ability to adapt to dynamic environments.

In this paper, we introduce a novel trust-based information retrieval mechanism, which is based on Ant Colony Optimization (ACO) algorithm. The ACO intelligent algorithm was used to collect and compute the trust value of each peer. Because of the dynamic feature of the ACO algorithm, our new P2P platform can adapt to the dynamic environments easily. The paper structure as follows: in the second section, we introduce the traditional trust-based P2P system; in the third section, we introduce the basic ACO algorithm and the task similarity concept; in the four section, we introduce the novel mechanism proposed; in the last section, we give a detailed experiment to illustrate the efficiency of our new algorithm of information retrieval.

2. Trust-based P2P Systems

Many researchers have aimed to find a more flexible algorithm to compute the trust value of the peer in the P2P network to distinguish the malicious peer and the good peer. EigenTrust [1] is the good algorithm to represent the trust-based P2P platform, which includes trust and recommendation. There exists a certain server to maintain the trust value of each peer in the platform. When a client peer wants to request a resource, he firstly sends the request information to all neighbors or the selected Top-*k* neighbors. After the candidate servers return the information about who maintains the required resource, the client peer should send a trust value request to the trust server to query the trust information about the entire candidate servers. At last, the client peer will select the server peer with the highest trust value to provide service. However, the main disadvantage of EigenTrust is the consumption of large network bandwidth. In order to get the trust value of the candidate server peer, the client peer must send a request message to the trust server in each information retrieval process.

Michlmayr [2] gives us a new routing mechanism based on ant colony optimization (ACO), which focuses on content-based search in P2P networks. Adler [3] proposed an optimal peer selection solution for minimizing the download delay subject to a budget constraint. In this study, we extended the algorithm referred in [2] to make it adaptive to selecting service peers in P2P networks, and realize the new algorithm on the Query Cycle Simulator [4].

3. ACO algorithm and task similarity

3.1 ACO algorithm

ACO is a technology which is used to find the most optimized path in a graph. It is introduced by Doctor Marco Dorigo in his Doctor thesis in 1992. ACO is one kind of swarm intelligent algorithm [5], which has been verified excellent in solving such problem that many simple peers to construct a complex system. The main steps for the ACO algorithm as follows:

Step1: Initialize the pheromone trail on every point;

Step 2: For every capable points to be selected, the probability is computed as following formulation:

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{r\in allowed_k}\tau_{ir}^{\alpha}(t)\eta_{ir}^{\beta}(t)}, & j\in allowed_k\\ 0, & otherwise\end{cases}$$

Step 3: After getting the required information, the pheromone trial on every point will be updated follows the formulation:

$$\tau_{ij}(t+1)=\rho\cdot\tau_{ij}(t)+\Delta\tau_{ij}$$

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{\ k}$$

Fig.1. the main steps of the ACO algorithm

3.2. Task similarity

Many literatures have proven that a peer may have some certain interests; it will concern the interesting thing in the special scope. Therefore, a server peer can complete a certain task in its interesting scope, and has none interesting about the scope out of its control. This pheromone we called task similarity. Supposed that there are two tasks named t_1 and t_2, the formulation to compute the task similarity between the two tasks as follows:

$$sim(t,t')=\{\omega\times sim_v(t,t')+\xi\times sim_f(t,t')+\lambda\times sim_t(t,t')\}$$

Where $sim_v(t,t')$ means the download speed similarity between two tasks, and $sim_v(t,t')$ can be described as follows:

(1)

$$sim_v(t,t') = \begin{cases} 0 & |T_{t1} - T_{t2}| > \varepsilon \\ e^{-|T_{t1} - T_{t2}|} & otherwise \end{cases},$$

Where ε is a very low value; T_{t1} is the time for the peer t_1 to download the given resource.

$sim_f(t,t')$ is the resource similarity between the two peers, which can be described as follows:

(2) $$sim_f(t,t') = \frac{\sum_{i=1}^{m} c_{ti} \times c_{t'i}}{\sqrt{(\sum_{i=1}^{m} c_{ti}^2)(\sum_{i=1}^{m} c_{t'i}^2)}}$$

Where c_{ti} represents the resources that the peer *t* has;

(3)

$$sim_t(t,t') = \begin{cases} 0 & |Tr_{t1} - Tr_{t2}| > \varepsilon \\ e^{-|Tr_{t1} - Tr_{t2}|} & otherwise \end{cases}$$

Where Tr_{t1} represents the trust value of the peer *t*;

(4) ω represents the weight of the $sim_v(t,t')$ in the task similarity; ξ represents the weight of the $sim_f(t,t')$; λ represents the weight of the $sim_t(t,t')$. $\omega + \xi + \lambda = 1$.

4. The intelligent trust-based platform

4.1. The main components

The intelligent trust-based P2P platform contains several components as follows:

(1) Trust collection component

We use the ACO algorithm to collect the direct trust and recommendation trust during the resource request and download process. The detailed collection process as illustrated in Fig.2.

(2) Request message routing component

The routing algorithm for every request message is the key issue in the information retrieval mechanism. Here, we use three tables as follows:

- Trust Value Table (TVT)

We use a trust value table contains the pheromone trail for every other peers. In the route table, the row represents each peer around the current peer, and the column represents the trust value for the corresponding peer.

- Service Time Table (STT)

The service time table is used to compute the approximate service time for the corresponding peer. The peer with the higher service time value will provide better service to the client peer.

- Content Similarity Table (CST)

The element in the content similarity table is computed based on the content similarity between the corresponding two peers. A higher content similarity value means the two corresponding peers have more similarity and the request will be completed with higher probability.

(3) Trust value update component

After the required resources were found, the trust value of the service provider will be added a certain positive value, and the trust valued of the malicious peer will be added a certain negative value.

4.2. The work flow

Step1: assign an initial trust value to each peer in the system;
Step2: create the three tables for every peer;
Step3: for the server peers, waitting for the request message; for the client peers, issues a request message for downloading a file;
Step4: A new ant named *F*ant was created just after the request message created by the client peer. The *F*ant select *k* neighbors to send the request message. The selection rule based on the three table content, that is, the selection route is based on the trust value of the destination peer, the content similarity between them and the time similarity between them. After computing the task similarity of all neighbors, the *F*ant will select the Top-*k* neighbors to relay the request message. The process will go on until the TTL (Time to Live) of the message is zero or the peer with the resoure was found.
Step5: When the peer with the required resource was found, the *F*ant will be destroyed, and a new ant named *B*ant was created. The *B*ant will come back on the same way, and adjust the value of the three tables for each peer on the way.

Fig.2. the work flow of our algorithm

5. The experiment result

We have coded the intelligent trust-based algorithm into a P2P experiment platform using the JAVA language.

The experiment we have made is to compute the hit rate difference between the trust-based intelligent platform and the platform without considering the trust value. From Fig 4, we can see that during the initialization phrase of the system, there exists slight difference between the two algorithms. We know that, when all peers enter into the system, they have not trade with any peer, so, the trust value of the others was the same value. Therefore, at cycle 1, the hit rate was very low for the two algorithms. With time go on, there are many trades among all peers, the trust value was aggregated and affects the trade behavior of the next trade. With this useful information, a peer can find a better server in the trust-based platform, so, the hit rate in the trust-based platform increase deeply.

6. Conclusion

To accelerate the information retrieval process and at the same time decrease the network bandwidth is the key issues in many networks especially for the peer-to-peer networks. This paper proposed a novel intelligent trust-based algorithm to collect the trust value, update the trust value and use the trust value to improve the information retrieval process. The future work was to improve the network balance of the platform and make the system more robust.

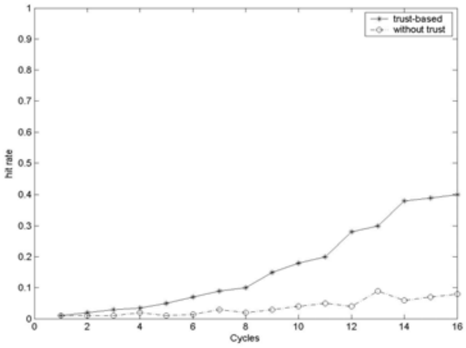

Fig.4. the hit rate comparison

7. Acknowledgement

This research was partially supported by grants 2006AA01Z455 from the National High-Tech Research and Development Plan of China, 60874075, 70871065 from National Science Found from China, 2004ZX17 and 2004ZX14 from the Natural Science Foundation of Shandong Province, J08LJ20 from Science Research and Development of Provincial Department of Public Education of Shandong and X061015 from the Science Foundation of Liaocheng University.

References

[1] Kamvar SD, and Schlosser MT, “EigenRep: Reputation management in P2P networks”, In: Proc. of the 12th Int'l World Wide Web Conf, ACM Press, New York, 2003, pp. 123-134.

[2] Elke Michlmayr, Arno Pany, and Gerti Kappel, “Using taxonomies for content-based routing with ants”, Computer Networks . 2007, Vol. 51, No. 16, pp. 4514-4528.

[3] Micah Adler, Rakesh Kumar, Keith W. Ross, Dan Rubenstein, Torsten Suel, and David D. Yao. “Optimal peer selection for P2P downloading and streaming”, In: Proc. of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), IEEE Press, Miami, Florida, USA, 2005, pp. 1538-1549.

[4] Schlosser M, Condie T, and Kamvar S, “Simulating a file-sharing P2P networks”, In: Proc. of the 1st Workshop on Semantics in P2P and Grid Computing (SemPGRID 2003), LNCS, Springer, Budapest, 2003, pp. 69-80.

[5] Dorigo, M., Gambardella, and L. M, “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”, IEEE Transactions on Evolutionary Computation, 1997, Vol. 1, No. 1, pp. 53-66.