Study of Two-level P2P Model on Self-adaptive Dynamic Network

Feixue Huang

Department of Economics, Dalian University of Technology, Dalian 116024, China

Tel: 86-411-8470-7210 E-mail:software666@163.com

Zhijie Li

School of Computer Science & Engineering, Dalian Nationalities University, Dalian 116600, China

Tel: 86-411-8765-6179 E-mail:lizhijie@dlnu.edu.cn

Abstract

This study's objective was to solve the problem that the structured P2P model was not appropriate for dynamic network. A self-adaptive dynamic P2P model of two-level is proposed. Super peers compose to a self-adaptive Distribute Hash Table network on the top layer, and lower ordinary peers take super peer in the same group as their center server to form a cluster. When super peer join or leave, it does not maintain the consistency of entire network's logic topology. Only updating routing table of its predecessor and successor, super peers update their routing table cache when they transmit messages. The experimental results indicate that if k is average amount of peers in each group, then search in self-adaptive P2P model of two-level only brings 1/k hops compared to Chord of same net size. When peer join or leave it produce messages at constants level. The conclusion indicates that the model could be more applicable in ad hoc P2P network with high churn rate compared with Chord network.

Keywords: Peer-to-Peer(P2P), Fault tolerance, Distribute hash table, Ad hoc P2P

1. Introduction

Based on the new computing infrastructure—grid technology(LI Z J,2009) (OLESZKIEWICZ J,2006): such as, Zhu G H(2004) discussed environment of China's knowledge grid; Li M L(2006) built a grid project of Shanghai's information services; WANG Y f (2007) built a grid framework for testing the engine. this thesis focuses on how to optimize the grid resource allocation using economic theory. Thus, resource management is a complex process (BERTEN V,2006). All existing P2P model can be classify to three categories by network logic topology: Centralized model, Decentralized unstructured model and Decentralized structured model. However such rigid structured model was not appropriate for dynamic network as it brings large number of messages when peer churns. Decentralized unstructured model is ineffective for its flooding search mechanism, and Decentralized structured model based on DHT (Distribute Hash Table) enjoys great advantage of simplicity and extensibility. CHEN H h(2006) proposed Semre X, a Semantic Similarity Based P2P Overlay Networks. In fact, P2P networks are focused on the sharing of resources (LING B,2005)(DOU W,2004)(WU K,2006)(LI Z Y,2006).Currently, Consistent hashing and DHT (Distribute Hash Table are often used in P2P networks, such as Chord (STOICA I ,2003), CAN (RATSANAMY S,2001), Tapestry(ZHAO B Y,2004), etc.

To address the above issues, a dynamic network which is more suitable for adaptive two-level P2P model is proposed. The model is organized into an adaptive DHT networks for the upper super-peer, and the organizational structure of lower ordinary peer is similar to Napster(SAROIU S,2003), which take the super-peers in the present group as the central directory server to form a cluster. The data actually is stored in an ordinary peer, and the super peers save data and deal with the requests from ordinary peers. When peers join or leave, the upper super-peer networks temporarily do not to maintain the consistency of the networks logical topology, only update routing table of its predecessor and successor, and the super peers update their routing table cache when they transmit messages. The lower peers retain the backup of super-peer; and generate a new super-peer when the super-peer fails. Adaptive two-level P2P system model has a better search performance, one query produces O(log N) (N is number of super peers)hops, and when peers join or left, it take relatively lower cost to maintain the networks topology, also has a certain degree of failure recovery mechanism. As the networks operate, it voluntarily adjusts load-balance and selects the high-capacity peer as the super-peer, so that the overall structure tended to be optimized.

2. Variables and principle of Model

Data format used in the system is similar with the general DHT Networks, is (key, value), key is the keywords corresponding to the data, key and the upper super-peer identifier (ID) are all m-bit identifier distributed by the consistent hash function. System provides function of the releasing the value, as well as inquiring the corresponding value of a given key. The format of Data Index is (key, routing): key is for corresponding keywords to the data, routing is for data storage of peer routing information. The overall structure of the system shown in Figure 1: System will voluntarily select the high-capacity peer as the super-peer. The upper super-peer networks are similar to Chord, the peer ID uniformly distributed in the identifier space, and form a ring in accordance with the peer ID.

The value of key is the index to the key value of *K*, stored in the peer whose first ID is equal to *K*, or a successor to *K* in the ring, namely the first encountered peer *K* start from the ring in clockwise direction. This peer becomes the successor of key. In addition, super-peer saved M-bit Rooting Table (becoming the Finger Table) to speed up the query. The difference with the Chord is that: the upper super-peer networks did not timely update the networks and synchronize the other peers in the Finger Table to reflect changes in the peer when peers join and leave, that is, the system allows the networks topology being inconsistent with the logic topology, at the same time only to update the peer's predecessor and successor peers, and to ensure the correctness of ring connectivity. And then when message transmitted between the super peers, each peer dynamically adjust the local Finger Table according to sources, so that the networks logical topology tends to be consistent. Also, the upper peers keep all routing information of ordinary peers in this group.

Lower ordinary peers take super peer in the same group as their center server to form a cluster, and the ordinary peers only keep routing information of upper peers in this group. Ordinary peer (key, routing) sent data index stored in the local to super-peers of the group, then according to the rules super-peer posted the index of the key value to a peer in the ring, namely, the successor to the key .When query, an ordinary peer transmits a query message to super-peer, and the super-peer only transmit the message on the super-peer ring, until finding the index of the data looked for, it will return the request message to the query peer, and then query peer voluntarily connect the destination peer to get data.

In each group, the super-peer select a ordinary peer in this group (does not contain itself) as a backup peer in accordance with peer ability, backup peer cache super-peer information, which include: the index to the key value, routing information of ordinary peers in this group, the super peer ID, the successor peer on the upper ring, predecessor peer. Backup peer can generate a new super-peer when super-peer invalid. To balance the load of super-peer in the system, system sets the minimum and maximum number of peers for each group. When number of ordinary peers under a super-peer exceeds the maximum, they split into two new groups. When number of normal peers is less than the minimum number of peers, the group will voluntarily merge into successor peer group of the super-peer on the Chord ring.

3. Algorithm Description

3.1 Data distribution and data query

The process of peer to publish data is described as follows:

1) The key value data to be released is calculated by consistent hash function.

2) To generate data indexing, such as (key, peer routing).

3) Sent to super-peer of this group.

4) The super-peer sent the index to the successor super-peer of the key value to save.

The search algorithm of peer is described as follows:

1) The value of the key data to be inquired is calculated by consistent hash function.

2) Transmit the query request of key value to super-peer of the group.

3) The super-peer transmits to the key's successor peer.

4) The peer returns the message, which contains the corresponding index of the key.

5) The query peer voluntarily connects to data- storing peer.

3.2 Algorithm of Peers' joining and leaving

Peers will issue a join request to the upper super-peer in the system when joining, the super-peer process the request. If number of ordinary peers under a super-peer in the group reaches the maximum the system allows, it splits a new group. This would avoid that the group is too great and the super-peers are overloaded and can make system more robustness. After received returning message of the request, ordinary peers obtain routing information from the super-peer, and are set as group servers routing that means it join the networks. The flow that Super-peer processes the leaving request: peers sent leaving request to super peer of the group when leaving the networks, super peer processes the request. If number of ordinary peers under a super-peer in the group reaches the minimum allowed by the system, the group will voluntarily merge to successor peer group of the super-peer on the Chord ring. This will avoid too many super-peers caused by too many small groups in the system.

3.3 Recovery Algorithm of Peer Failure

A variety of operation in system mainly carried out on the upper super-peer, ordinary peer only depends on the super-peer. It can be seen that ordinary peer failure does not produce any effect on the entire networks topology, at most when searching for the early release of data index of the failed peer, it fails when other peers acquire data. The operation of the entire networks will not be affected, so here only the super-peer failure recovery mechanism is considered.

Each group in the system has an ordinary peer to back up certain essential information of super-peer, including: index of the key value cached by super peer, routing information of ordinary peer, super-peer ID, successor peer of super-peer in the upper ring, routing information of precursor peer. Therefore, when a super-peer fails, the backup peer has enough information to restore a new super-peer.

In order to maintain the effectiveness of the backup peer information, ,when the above information of super-peer changes, the backup peer must simultaneously be updated, and when there are ordinary peers joining the group, the super-peer select the ordinary peer of the largest capacity for backup.

The backup peer takes the way of interval of time period checking to monitor super-peer, once a super-peer is not connected, the backup peer transform into a new super-peer, while selecting a new backup peer in the group.

The algorithm of recovery process of backup peer is described as follows:

1) Backup peer transform into a super-peer.

2) Add the backup information to itself.

3) Inform the super-peer on Chord ring to change the successor peer into its own route, and inform successor peer to change its predecessor into its own route.

4) Inform the ordinary peer within the group update the super-peers routing into itself.

3.4 Analysis of System performance

In Chord model, each peer store O (log N) routing information of other peers, a query hops are kept at 0(log N), N is the networks size. Table 1 lists the comparison of hops for self-adaptive Chord and Chord.

When peers join or leave, due to update the other peers in the networks, regardless of under what circumstances Chord need launch at least log N queries, and a query generate O (log N) hops, it totally has generated O ($\log^2 N$) hops. As the super-peer of upper ring temporarily stops real-time maintenance of networks, a major operation of Self-Adaptive Chord at this point is to find the predecessor peer to be joined, and then transmit two messages. Therefore, in the best case, the peer requests its predecessor to join the networks, peers produced only two hops. The worst and the general cases result in O (log N) hop. Compared to Chord, Self-Adaptive Chord can effectively reduce the number of messages caused by the peers join or leave, at the same time as the networks operates, the query efficiency become closer to Chord.

When query, in the best circumstances, when a query peer key is stored in constant successor peers after the initiated peer, Chord and Self-Adaptive Chord only produce constant hops. In the worst case, if the query key's precursor is the query peer, Chord needs to transmit message among log N peers to the destination peer.

If all the peers of the Self-Adaptive Chord are newly joined the networks, and they are all directly request their predecessor peer to join, then all peers only have successor peer routing and self-routing in the Finger Table, so the peer will traverse to find the destination peer along the ring through its successor peer, resulting in N hops. Under general case, because Self-Adaptive Chord will constantly adjust the networks logical topology, a query hops of Chord and Self-Adaptive Chord both is O (log N).

Self-Adaptive Chord is only the upper layer of adaptive P2P mode, the networks size increase significantly compared to Chord, making n as the average number of peers in each group, if the efficiency of P2P model is similar to Chord, the networks size increases by n times; In addition, as the system ensures that the number of ordinary peers of each group is kept with a certain range, so it not only avoids the excessive group creating a overload super-peer, or a large area networks paralyzed after a single peer failure, but also avoids too many groups with too few peers and purposefully limit the number of super-peers, so that system performance can be further enhanced.

4. Experiments

4.1 Comparison of query efficiency between Adaptive two-level P2P model and Chord model

The experimental procedures are the new 500-peer in Chord networks and sequential implementation of the 20 Sections respectively in 500 super-peer adaptive two-level P2P model, 500 queries in each section.

As the results shown in Figure 2, the query implementation efficiency of Chord is more stable, and every 500 queries generated more than 3,000 messages, with average 6 messages each query, this is because the number of messages a Chord query produces is O (log N). But the adaptive two-level P2P model's efficiency of query is worse than Chord in the first groups, because in the newly generated networks, the Finger Table of most peers is initialized in relation to its successor, these peers do not know the whole the logic of the networks topology, and is likely to transverse the entire ring peer by peer by the successor peer, therefore less efficient than Chord. But, after some inquiries, the peer updates local routing information by transmitting the message dynamically, and we can see the more lately implemented the query is, the more closer the number of messages is to Chord. At this point in adaptive P2P networks model the number of all the peers is about 4000, showing that when the size of two-level P2P model adaptive networks is approximately 8 times of Chord, its query efficiency is still close to Chord.

4.2 Analysis of the distribution of the peer capacity in Adaptive two-level P2P model

The purpose of this experiment was to analyze the distribution of the peer capacity in Adaptive two-level P2P model. Experimental procedures is to generate a network with 5000 peers, capacity of each peer is a random number between 1 and 10. Then check distribution of the super peer capacity.

As the results shown in Figure 3, the capacity of most super-peers is a value around 8,9,10. It can be seen that Adaptive two-level model P2P optimize network's structure, dynamically adjust networks structure, and select the peer of large capacity as the super peer.

5. Conclusion

The experimental results indicate that if k is average amount of peers in each group, then search in self-adaptive P2P model of two-level only brings 1/k hops compare to Chord of same net size, when peers join or leave it produce messages at constants level. It also has effective mechanism for fault tolerance. The conclusion indicates that the model can be more applicable in ad hoc P2P network with high churn rate.

Acknowledgments

This work is supported by the funds project under the Ministry of Education of the PRC for young people who are devoted to the researches of humanities and social sciences under Grant No. 09YJC790025; software + X research of Dalian University of Technology under Grant No. 842301.

References

BERTEN V, GOOSSENS J, JEANNOT E. (2006). on the distribution of sequential jobs in random brokering for heterogeneous computational grids. *IEEE Transactions on Parallel and Distributed Systems*, 17 (2): 113 – 124.

CHEN H h, JIN H, NING X m, et al. (2006). SemreX: A Semantic Similarity Based P2P Overlay Networks. *Journal of Software*, 17(5):1170-1181.

DOU W ,WANG H M, JIA Y, et al. (2004). A Recommendation-Based Peer-to-Peer Trust Model. *Journal of Software*, 15(4):571-583.

LI M L, WU M Y, LI Y, et al. (2006). ShanghaiGrid: an Information Service Grid. *Concurrency and Computation: Practice & Experience*, 18(1): 111-135.

LI Z J, CHENG C T. (2009). An Evolutionary Game Algorithm for Grid Resource Allocation under Bounded Rationality. Concurrency and Computation: Practice and Experience, 21(9): 1205-1223.

LI Z J, CHENG C T, HUANG F X. (2009). Utility-driven solution for optimal resource allocation in computational grid. *Computer Languages, Systems and Structures*, 35(4): 406-421.

LI Z Y, XIE G G, MIN Y H, et al. (2006). Topology Matching Method for Structured P2P Systems. *JOURNAL OF SYSTEM SIMULATION*, 18(5):1181-1185.

LING B, WANG X y, ZHOU A y, et al. (2005). A Collaborative Web Caching System Based on Peer-to-Peer Architecture. *CHINESE JOURNAL OF COMPUTERS*, 28(2):170-178.

OLESZKIEWICZ J, LI X, LIU Y H. (2006). Effectively Utilizing Global Cluster Memory for Large Data-Intensive Parallel Programs. *IEEE Transactions on Parallel and Distributed Systems*, 17(1): 66-77.

RATSANAMY S, FRANCIS P, HANDLEY M, et al. (2001). A scalable content-addressable network. *ACM SIGCOMM Computer Communication Review*, 31 (4): 161 - 172.

SAROIU S, GUMMADI K P, GRIBBLE S D. (2003). Measuring and analyzing the characteristics of Napster and Gnutella hosts. *Multimedia Systems*, 9 (2): 170 - 184.

STOICA I, MORRIS R, LIBEN-NOSTRONG D, et al. (2003). Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, 11 (1): 17-32.

WANG Y f, DONG X s, HE X q, et al. (2007). Grid Testing Engine:a Infrastructure for Constructing Grid Testing Environment. *JOURNAL OF XI'AN JIAOTONG UNIVERSITY*, 41(8):884-888.

WU K, DAI H, YE B l, et al. (2006). Research on NS2-Based P2P Networks Simulation Platform. *JOURNAL OF SYSTEM SIMULATION*, 18(8):2152-2157, 2169.

ZHAO B Y, HUANG L, STRIBLING J, et al. (2004). Tapestry: A Resilient Global-scale Overlay for Service Deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41-53.

ZHUGE H. (2004). China's e-science knowledge grid environment. *IEEE Intelligent Systems*, 19 (1): 13-17.

Table 1. Comparison of hops for self-adaptive Chord and Chord

Peer Operation	networks	Chord	Self-Adaptive Chord
inquire	Best case	O(1)	O(1)
	worst case	O(log *N*)	O(*N*)
	General case	O(log *N*)	O(log *N*)
join or leave	Best case	O(log^2*N*)	O(1)
	worst case	O(log^2*N*)	O(log *N*)
	General case	O(log^2*N*)	O(log *N*)

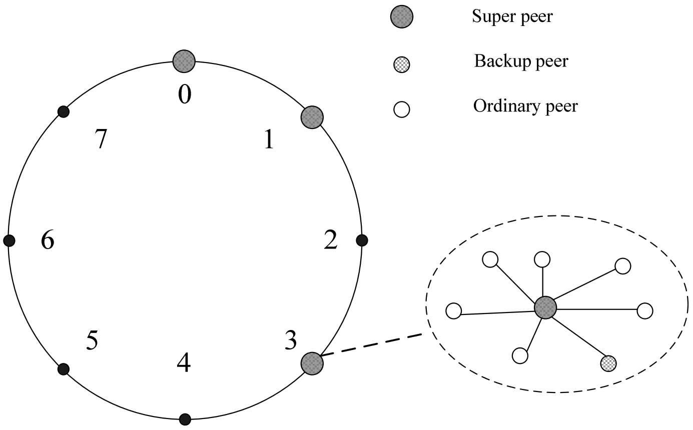

Figure 1. The overall structure of the system

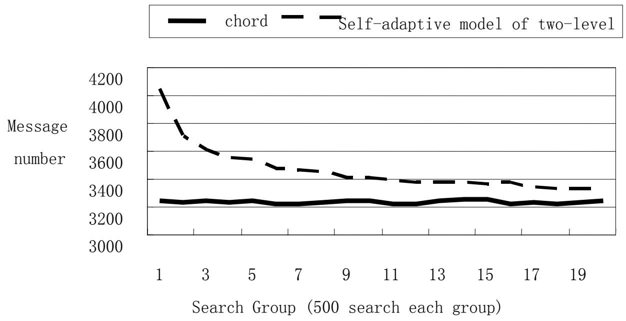

Figure 2. Comparison of query performance for self-adaptive model of two-level and Chord

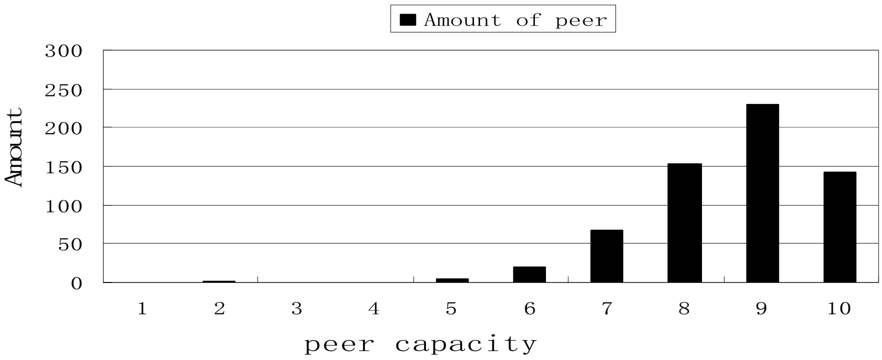

Figure 3. The ability distribution of super-peer for self-adaptive model of two-level