EAI Endorsed Transactions
on Internet of Things

Research Article EAI.EU

A Search Algorithm Based on K-Weighted Search Tree

Lianhai Yuan [1],*, Xiangwen Li [2] and Lin Zhou [3]

[1] Department of Electronic information and Computer Engineering, the Engineering &technical College of Chengdu University of Technology Leshan, China
[2] Department of Electronic information and Computer Engineering, the Engineering &technical College of Chengdu University of Technology Leshan, China
[3] Department of Electronic information and Computer Engineering, the Engineering &technical College of Chengdu University of Technology Leshan, China

Abstract

Aiming at the issue of low efficiency in Peer-to-Peer (P2P) network system, a search algorithm based on K-weighted search tree is proposed. The k-weighted search tree serving the search is constructed. The nodes are ranked from top to bottom in the tree according to the query hit rate, and the nodes with large hit rate and stable are on the tree layer, the search can thus determine the direction of the message diffusion. By caching the upper node, establishment of search results, using node index, overheated resource replication and add remote neighbours for leaf node, and other methods to improve search efficiency and balance load. The analysis and simulation results show that the proposed algorithm can greatly reduce the invalid message with higher search efficiency, and maintenance of the search tree is less expensive.

Keywords: Searching algorithm, k-weighted search tree, Peer-to-Peer.

Received on 18 February 2017, accepted on 05 April 2017, published on 14 April 2017

doi: 10.4108/eai.15-1-2018.154108

1. Introduction

Unstructured P2P network systems have been widely used with its simple structure, easy organization in large scale network resources sharing. The search efficiency of resource determines the availability of such system, and key to the system. Blind flood is the initial and basic search method for unstructured P2P, but overhead is too large, resulting in poor system scalability. Random walker search is another basic search method in unstructured P2P.Single walker mode cannot be practical because of its low efficiency and too long delay.

Plane P2P structure is divided into multiple layers, the upper node management part of the lower nodes, and the same layer nodes connected each other. Such as KaZaA using super-section, to a certain extent, ease the flow problem, but the super-node may cause a single point of failure within the region and become a bottleneck in system performance.

The purpose of classification resources and nodes is to narrow the search, to speed up the search speed, classification can be based on geographical, unit, interest or content.P2P nodes are divided into multiple groups, while each node belongs to different classes, each class form a relatively independent logical network, the search message only with the search content related to the proliferation of logical networks, thus reducing the scope of the search to a certain extent reduce the amount of redundant traffic . Hierarchical and taxonomic methods used in combination, the nodes are divided into super peer, deletegate and local nodes, According to the theme of the network hierarchical classification, different types of nodes distributed in different layers.

Selectively routing can make the search direction, purpose of the choice of the basis of the search results can be the index, the content between nodes stored between

*Corresponding author. Email:419129498@qq.com the phases like, other historical information. Part of the routing index method, the routing index only maintain the most popular request content index, the other content is still using the traditional diffusion method. The probability routing table is used to maintain the neighbor information, according to the route table; the search request can reach the destination node with a higher probability and a shorter path. Some researcher use a grouping and guidance request based on interest forward to reduce bandwidth consumption.

Caching and multiple replica policies can improve system scalability. Combine a uniform and proportionate resource replication method to reduce the search range by controlling the placement of copies, depending on the requested ratio, the distribution ratio of the fixed copy.

Above P2P system proposed a variety of ways to improve the efficiency of search, but there are still repeated messages, search efficiency is low or delay and other issues. In this paper, we propose an unstructured P2P search model based on the K search tree. The model constructs a search structure based on the K-tree, and adopts the lifting and lowering mechanism. The position of each node in the tree is arranged according to the query hit rate, and the node with the larger weight in the upper layer of the tree, so that the nodes in the system are regularly distributed, the more the upper section of the festival, the greater the probability of having hotspots. And the use of the establishment of search results historical index, and the search initiates a node index, caches the upper node, partially overrides the resource copy and is leaf nodes to increase the remote neighbors and other methods to further improve the search efficiency and balance negative contained. Analysis and simulation results show that we can significantly reduce the inefficient flow and increase search efficiency, and the search tree maintenance overhead is very small.

2. System Model

The system topology of unstructured P2P is a logical network with a power law distribution and a high degree of aggregation, called the original network. In the maintenance system structure integrity, reliability and fault tolerance, the original network has good performance, so still retain the existence of the original network. In the tree structure, the neighbor of the node has only the parent, the child node and the very small number of redundant nodes, the node degree is small, the system structure maintenance cost is very small, and the duplicate message is not generated when the search request is spread. And the original network based on the search efficiency is low, inefficient flow, so the original network to build a K-tree for resource search. Further optimize the search properties of the tree, assign weights to each node in the tree, and give the node a weight regularly arranged, the K weighted search tree is obtained.

Definition 1: a tree is K weighted search tree if the conditions are met:

(i) There are at most K child nodes per node;

(ii) Only the bottom and sub-layers of the tree have leaves and nodes with fewer nodes than K nodes;

(iii) For any node, its weight is less than or equal to the weight of the parent node, and greater than or equal to the weight of all the children

Assuming that the system has N nodes, the node i is denoted as Ni, the child node set of Ni are denoted as S(i), the brother node set of Ni are denoted as B(i) and F(i) represents the father node set. In order to ensure the reliability and fault tolerance of the search tree structure, there is a small number of backup nodes in each node except the parent and child nodes. The root node and the first node are defined as the backup nodes, and the backup nodes of the other nodes are the grandfather nodes. The search tree in the initialization and reconstruction process may not satisfy the condition (3) of definition 1, but still called K-weighted search tree.

Initialize the construction of the search tree algorithm

```
ConstructTree (newPeer, tempPeer)
{
      if newPeer is the first Peer then
      root = newPeer
      return TRUE
      end if
      if subTreeFull(tempPeer) is TRUE or sonNum(tempPeer)==0 then // is leaf or is full
      tempPeer = fahter(tempPeer)
      while subTreeFull(tempPeer)== True
      {
         tempPeer = fahter(tempPeer)// The subtree is full and up
      }
      end i f
      while sonNum(tempPeer)== k
      {
      if sonPeer ∈ son(tempPeer) && subTreeFu ll(sonPeer)! = TRUE && sonNum(sonPeer)>0 then
               tmpPeer = sonPeer
      end if
     }
     newPeer add in tempPeer
     return TRUE
}
```

Related term: newPeer: new arriving node; tempPeer: any node that already exists in the tree;

SonNum(Ci): Calculates the number of child nodes of Ci; subTreeFull (Ci): whether the subtree with Ci is the full tree.

3. Search model

The historical query hit rate is the weight of the node in P2P search tree, when the system is knotted the structure tends to be stable, from the top of the tree down, the node hit rate and stability gradually reduced small, the upper nodes of the hot spots more resources, while the lower non-hot resources, because the more, This provides a more accurate source of hot and cold distribution. Any node the direction of the hotspot area can be easily determined and can be reached in a very short time up to for nodes with non-hotspot resources, use remote connections to enhance them the relationship between the same can also improve the efficiency of unpopular resources search. While taking parts hotspot resource replication, indexing and upper node caching mechanisms to further improve the search efficiency at the same time, to solve the tree structure caused by the part of the node load heavier happening. New nodes are added from the lower level, frequent access to the system, the total instability of the node is at the lower level of the tree, so the stability of the system search performance can be guaranteed certificate.

The weight of the node is the hit rate (HR, Hit Ratio),HR from the node to join the system to the current moment of the query hit rate comprehensive evaluation value, reflecting the node has the resources in the network heat.

Set the node arrives to the system time is 0, each time period length τ. In the jth during the time period, the number of times to reach Ci is QueryNum, where the hit times the number is HitNum, and Ri (j) represents the hit rate of Ci in the jth period.

Ri (j) = HitNum / QueryNum

Wi (j) is Ci in the [0, jτ] time within the query hit rate, recursively defined as :

$Wi (j) = \alpha Wi (j -1) + (1 - \alpha) Ri (j), 0 <= \alpha <= 1, j> 0$

When j equals 0, the initial value Wi (0) is defined as

Wi (0) = 0

α is the hit rate coefficient, take different values, reflecting the query hit rate is more focused on long term or near-term results, the smaller α , the more recent the hit rate is more important. The latest query hit rate of Ci is simply expressed as Wi.

Proposition 1: for any node Ci, Wi (j) greater than or equal to 0, less than or equal to 1

Prove: the analytical equation Wi (j) can be obtained from the recursive equation

$$Wi (j) = (1 - \alpha) \sum_{k=0}^{j-1} \alpha\ j -1 -k\ Ri(k)$$

Since 0 <= α <= 1, 0<= Ri (k) <=1, when Ri (s) = 0, 0<= s <=j -1, Wi (j) takes the minimum value 0, and when Ri (s) = 1, 0<= s <= j -1, Wi (j) to obtain the maximum value

$Wi (j) = (1 - \alpha) \sum_{k=0}^{j-1} \alpha\ j -1 -k =1 - \alpha\ j$, available Wi (j) <= 1

So 0 <=Wi (j) <=1, proved.

Because the request is forwarded between neighbors of the nodes in the tree structure, the number of requests received in adjacent nodes is similar, so the neighboring nodes of the HR is comparable, HR can reflect the degree of node resource satisfaction with the request.

To satisfy the condition (3) in definition 1, when the HR value of the node changes or is made system initialization process, the need for nodes in the tree position to do the appropriate adjustment, transfer the whole process is called the lift-off process, corresponding to the RS (Rise and Sink) algorithm. Node from the current layer to the upper layer, is called the upgrade, and vice versa for the downgrade.

For Ci, the parent node is Cf, the sibling node Cb ∈ brother (i), the child node, Cs ∈ son (i), each node maintains a row containing the child node and its own list table (H RL, HR List). The RS algorithm is as follows.

Algorithm: Search tree rise and sink

Step1: If Cs finds the timer is changed to zero, the latest query Hit rate Ws is calculated, and through the message NEW-RATIO sent to Ci, and updated H RL, reset timer.

Step2: Ci received the NEW-RATIO message from Cs, take out the message Ws, and update H RL. View HR of all child nodes in HRL if Wi is still the maximum, then complete; otherwise carry out rise and sink. Assuming that HR of Cs is the maximum, then Send REQ-UPGRADE message to Cs, request rise and sink processing.

Step3: when Cs received the message REQ-UPGRADE, if agreed to upgrade, send message AGR-UPGRADE to Ci; otherwise send REJ - UPGRADE.

Step4: if Ci received REJ - UPGRADE, then complete; otherwise start to rise and sink operation: Cs with Cf as the new parent node, and with Ci and {son (i) -Cs} point as a child node, the original Cs child node to Ci for the new parent node. Rise and sink complete.

Algorithm Description: A node cannot be raised for two consecutive periods of time.

The rising and sinking is the process of gradually optimizing the search tree. When Ci is higher than the HR of the brothers and the parent node, Ci is exchanged with the parent node and is updated once. The nodes that are newly added to the search tree are always located at the bottom or sub-layer of the tree as leaf nodes. When the system is running for some time, the structure of the search tree tends to be relatively stable, stable and reliable, and the nodes with higher HR can always be located on the upper layer of the tree, stable but HR low nodes and unstable nodes with high HR, The relative middle layer of the tree is unstable and the node with low HR is located at the lower level of the tree.

4. K-weighted search tree algorithm

There is no available historical search experience data, or for other needs (such as non-hotspot resources) in the initial state of the system's operation or in the short term of the system, the request is forwarded to all neighbors, the nodes of the current node's fault-tolerant backup do not participate in the search. The search for a stop condition is TTL=0 or the search is successful or there is no node to forward. The basic search method does not generate duplicate messages, and the upper limit of the search delay (jump) is twice the tree height. With the increase of system running time, search tree gradually become orderly and relatively stable, the node has accumulated a certain search experience information, so the search algorithm can be optimized.

The basic search method does not generate duplicate messages, and the upper limit of the search delay (hop) is twice the tree height. With the increase of system running time, system search tree gradually become orderly and relatively stable, the node has accumulated certain search experience information, so the search algorithm can be optimized.

This paper uses a caching strategy based on the search results index to improve the search effectiveness. Based on the basic diffusion and caching index, a search algorithm is established.

```
Function SearchProcess(QueryReguest )
{
   destPeer = RetriveRIList (QueryReguest , RIList)
    if destPeer == Null Then
        destPeer=RetriveSPList(QueryReguest,SPLi st)
   endif
   UpdateSPList (QueryReguest , destPeer)
   if destPeer ! = Null
        SendTo(QueryReguest, des tPeer)
        UpdateRILi st(QueryReguest , destPeer)
        Modif yRecentHitRat io()// update latest HR
        return TRUE
   endif
   if RetrieveLocal(QueryReguest)! =NULL then
        SendTo(resultSet , sourcePeer)
        SendTo(resultSetAbs  t  ract  ,  lastPeer)//return
   value
        ModifyRecentHitRatio()
        return TRUE
   endif
   if TT L > 0 then // Forward request
   SendTo(QueryReguest  ,  all  neighbors  except
lastPeer)
   return FALSE ;
   endif
}
```

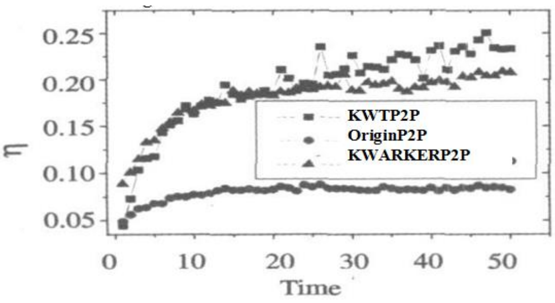

Figure 1. Search efficiency comparison

5. Conclusion

Compare with other search algorithm in peer to peer networks, the algorithm of this paper with higher search efficiency and smaller latency. The nodes are randomly distributed in the tree; the nodes are gradually sorted by the hit rate from top to bottom rules, for the originating node to receive the first return to the results of the time with the system changes. Figure 1 shows the performance of k-weighted search tree search algorithm. Algorithm has the smallest delay, because search direction is most clear, most of the message can quickly reach the hot area, thus improving the speed and probability of finding the results.

Acknowledgements.

In this paper, the research was sponsored by the Youth Fund Project of the Engineering &technical College of Chengdu University of Technology (Project No. 2016-YY-JG21).

References

[1] Kumar A , Xu J .Efficient and scalable query routing for unstructured peer-t o-peer network s .INFOCOM 2015

[2] Chen SyuYang , T Wen-Hsien , et al. A Multilayer Topic-Group based P2P Network .IEEE AINA , 2012

[3] Liu Yunhao , Liu Xiaom ei , et al. Location-aware topology matching in P2P systems .INFOCOM 2014 , 2014 .2220 ～ 2230

[4] Stephane B, et al. Exploiting local popularity to prune routing indices in peer-t o-peer system s .DEXA' 05, 2015

[5] Matei R, Iam nit chi A, et al. Mapping the Gnutella Network .IEEE Internet Computing , 2012 , 6 :50 ～ 57

[6] Leibowitz N , Ripeanu M , et al. Deconstructing the Kazaa network .In:WIAPP 2013 , 2013 .112 ～ 120

[7] Xin liu , et al. A category overlay infrastructure for peer-to-peer content search .In .IEEE IPDPS' 15, 2015

[8] Chen Wen-Tsuen , Chao Chi-Hong , et el. An Interested-based Architecture for Peer-to-Peer Network Systems .IEEE AINA, 2012

[9] Cohen E, S henker S .Replication strategies in unstructured peer to-peer networks .In Proceedings of ACM SIGCOMM' 12 , 2012

[10] Calvert K ,Zegura E .GT-ITM :Georgia Tech Internetwork topology models. Http: //www.cc.gat ech .edu/ f ac/ Ellen .Zegura/graphs .html

[11] Information on http://www.baidu.com