Computers and Electrical Engineering 38 (2012) 1456–1466

Contents lists available at SciVerse ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A dynamic load balancing strategy with the push and pull approaches in DHT networks ☆

Xianfu Meng *, Yalin Ding

School of Computer Science and Technology, Dalian University of Technology, No. 2, Linggong Road, Dalian 116024, China

ARTICLE INFO

Article history:
Received 13 May 2011
Received in revised form 18 April 2012
Accepted 18 April 2012
Available online 12 May 2012

ABSTRACT

The dynamic load imbalance problem, probably caused by the heavy-tailed distribution of file requests, negatively impacts on the distributed hash table (DHT) networks' availability. The existing solutions mainly employed the local load information to design the load balancing strategies, which often need to calculate the peers' loads and execute the balancing procedures periodically, and thus their effectiveness could not be guaranteed and network bandwidth is wasted. To address this problem, we first describe the mechanisms for managing the download volume and the upload volume of each peer, as well as the information of the heavily loaded nodes and the lightly loaded nodes classified by double thresholds, and then we present a novel load balancing strategy which transfers the loads from the heavily loaded nodes to the lightly loaded nodes with the push and pull approaches. The simulation results show that our scheme is effective and efficient in handling the load imbalance problem in DHT networks.

1. Introduction

As a novel distributed computing environment, peer-to-peer (P2P) networks are attracting more and more attention. Each node in P2P networks plays the both roles of receiving services as a client and providing services as a server, and thus the problem of single point of failure in the traditional client/server systems is eliminated. A P2P network provides an excellent platform with good scalability and fault tolerance for many distributed applications, such as file sharing systems, etc.

Distributed hash table (DHT), a kind of structured P2P network, has some advantages over unstructured P2P networks in terms of retrieval precision and computational complexity. However, it also suffers from some significant shortcomings: the high cost of churn resilience and the load imbalance [1], etc. Among those shortcomings, the load imbalance problem negatively impacts on the system response time and the network availability. Therefore, how to solve the load imbalance problem has been an important research focus in DHT networks [2].

The loads in DHT networks can be classified into two kinds, i.e. the static load and the dynamic load. The static load usually refers to the storage load, including the storage taken by the index information, the routing information and the shared files, which almost does not change with the upload and download actions. On the other hand, the dynamic load mainly refers to the load generated by the requests of download actions, including the loads of upload and download. Different from the static load, the dynamic load of a node negatively impacts on the network performance in terms of the retrieval efficiency and response time when it exceeds a certain threshold within a short time interval. In this paper, we only focus on the dynamic load balancing scheme.

☆ Reviews processed and approved for publication by Editor-in-Chief Dr. Manu Malek.
* Corresponding author. Tel.: +86 411 84706008.
E-mail address: xfmeng@dlut.edu.cn (X. Meng).

0045-7906/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compeleceng.2012.04.012

Since there is no centralized peer in a DHT network, it is difficult for any peer to obtain the load information about the whole network. For this reason, the existing researches either employ local load information to design load balancing strategies or inquire the heavily loaded nodes or lightly loaded nodes periodically, leading to the shortcomings in effectiveness of load balancing and the consumption of network bandwidth. This paper, from the global perspective, focuses on the load balancing strategy in a DHT file sharing system by taking advantages of the characteristics of the DHT network. Our main work is the following:

(1) Due to the lack of the mechanism for managing and retrieving the dynamic loads, the existing solutions either take the number of response messages as the dynamic load of a node, or assume a node heavily loaded or lightly loaded, based on which to tackle the load imbalance problem. It is, however, obviously unreasonable. This paper takes the upload volume and download volume of a node as its dynamic load, which accurately reflects the real dynamic load of a node. On account of the selfishness of the nodes in a P2P network, the dynamic load of a node is managed by its neighbor nodes, rather by itself.
(2) To avoid traversing all the nodes in the network to retrieve the heavily loaded nodes or the lightly loaded nodes, an effective and light weighted mechanism for managing and retrieving the information of those nodes is proposed by using the features of the DHT network. Since the heavily loaded nodes and lightly loaded nodes are divided by the dynamically adjustable double thresholds, only the node set with heavier loads and the node set with lighter loads are taken into account each time when we tackle the load imbalance problem. Therefore, it not only reduces the computational complexity and the waste of network bandwidth, but also makes the load balance more efficiently.
(3) We employ the push and pull approaches, triggered by the neighbor nodes of heavily loaded nodes and lightly loaded nodes respectively, to transfer the loads from heavily loaded nodes to lightly loaded ones. They not only tackle the load imbalance problem as soon as it happens, but also eliminate the non-cooperative problem brought by the nodes due to their selfishness for load transfer.

The remainder of the paper is organized as follows. Section 2 presents the related work on the load balancing strategies in P2P networks and Section 3 demonstrates several definitions; Section 4 details the dynamic load balancing scheme, including the collection and management of the dynamic load and the dynamic load transfer from the heavily loaded nodes to the lightly loaded nodes with the push and pull approaches; Section 5 discusses the settings of the relevant thresholds and the computational complexity, and Section 6 evaluates the effectiveness of our strategy. Finally, Section 7 concludes the paper and gives our future research focus.

2. Related work

As a kind of structured P2P network, DHT-based networks map the nodes and the resources to the same ID space by using hash operations to improve the efficiencies of retrieval and routing. Generally speaking, the difference in undertaking keys' space of each node, the difference in the number of sharing files of each node and the imbalance in information requests are the main causes of generating the load imbalance in DHT networks [3]. The main approaches proposed to tackle the load imbalance problem in DHT networks include file caching [3,4], fair ID space partitioning [5], virtual server [6–8] and multi-hashing [9]. We will discuss those strategies in the following.

Work in [3] mainly used the file caching approach to solve the load imbalance problem, which caches the frequently requested files held by a heavily loaded node on their retrieval paths. Since the load information has to be attached to the request messages, and also the nodes are required to calculate their request loads, routing loads and their in-degrees every T period, the approach significantly increases the network overhead and the nodes' computational complexity. Moreover, this approach takes the number of response messages as the node's dynamic load; obviously it does not suffice, as the dynamic loads generated by different request messages could be totally different.

The fair ID space partitioning is an approach which partitions the ID space according to the capabilities of the nodes [5]. This approach, however, is difficult to adapt to the changes of the dynamic loads as the ID partitioning approach can only be used for the static load. The virtual server approach is another one that balances the loads among the nodes by transferring the virtual servers from the heavy nodes to the light nodes [6]. In the one-to-one scheme, which is the simplest one, each light node can periodically pick a random ID and then perform a lookup operation to find the node that is responsible for that ID. If that node is a heavy node, then a transfer may take place between the two nodes. Apparently, each light node is blind and uncertain on the choice of the heavy node. Some improvements on Ref. [6] were made in the later researches. Researchers of Ref. [7] used a number of nodes as the directories to receive and store the load information, i.e. each directory collects load information from the nodes who contacted it. Every T seconds it computes a schedule of virtual server transfer among those nodes managed by that particular directory. Because each directory can only be responsible for a part of the nodes' load information in the network, it is difficult for an arbitrary directory to achieve an effective schedule for the entire network. Furthermore, the work in Ref. [7] did not describe the mechanism for managing and retrieving the dynamic load of a node. In addition, one of the main drawbacks of using virtual servers for balancing the load is that we must determine how many virtual servers a node should host. This is an obstacle to applying the virtual server based load balancing algorithms in real DHT networks.