



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

Load Balancing in Structured Peer to Peer Systems

DR.K.P.KALIYAMURTHI¹, D.PARAMESWARI²

Professor and Head, Dept. of IT, Bharath University, Chennai-600 073¹

Asst. Prof. (SG), Dept. of Computer Applications, Jerusalem College of Engg., Chennai-600 100²

ABSTRACT: Over the past few years, peer-to-peer (P2P) systems have rapidly grown in popularity and have become a dominant means for sharing resources. In these systems, load balancing is a key challenge because nodes are often heterogeneous. While several load-balancing schemes have been proposed in the literature, these solutions are typically ad hoc, heuristic based, and localized. In this paper, we present a general framework, HiGLOB, for global load balancing in structured P2P systems. Each node in HiGLOB has two key components: 1) a histogram manager maintains a histogram that reflects a global view of the distribution of the load in the system, and 2) a load-balancing manager that redistributes the load whenever the node becomes overloaded or under loaded. We exploit the routing metadata to partition the P2P network into no overlapping regions corresponding to the histogram buckets. We propose mechanisms to keep the cost of constructing and maintaining the histograms low. We further show that our scheme can control and bound the amount of load imbalance across the system. We propose a new framework, called Histogram-based Global Load Balancing (HiGLOB) to facilitate global load balancing in structured P2P systems. Each node P in HiGLOB has two key components. The first component is a histogram manager that maintains a histogram that reflects a global view of the distribution of the load in the system. The histogram stores statistical information that characterizes the average load of no overlapping groups of nodes in the P2P network. These nodes are connected to P through its neighbor nodes. The histogram information can be used for two purposes. On one hand, it is used to determine if a node is normally loaded, overloaded, or under loaded. On the other hand, it is used to facilitate the discovery of a lightly loaded node or a heavily loaded node for the load-balancing process when it is needed. The second component of the system is a load-balancing manager that takes actions to redistribute the load whenever a node becomes overloaded or under loaded.

I. INTRODUCTION

PEER-TO-PEER (P2P) systems have emerged as an appealing solution for sharing and locating resources over the Internet. Several P2P systems have been successfully deployed for a wide range of applications. The basic approach to load balancing is to find a pair of nodes—one that is heavily loaded and the other lightly loaded—and redistribute the load across these two nodes. However, it is far from trivial to (globally) balance the load in a P2P system. There are two main issues in P2P's load balancing: 1) how to determine if a node is overloaded or under loaded, and 2) if so, how to find a suitable partner node with which to redistribute the load. A popular solution is to let each node in the system query for the load of an arbitrary number of other nodes periodically. If the number of queried nodes is large enough, the node can approximate the average load of the system, and hence, it can determine if it is overloaded or under loaded. If the node is overloaded (or under loaded), it redistributes its load with the queried node having the lightest (or heaviest) load since that node should be a lightly (or heavily) loaded node. The main problem with this method is that it can only guarantee the global load balance of the system with some probability. On the other hand, we can use a separate DHT such as Skip Graph to maintain the nodes' load distribution.

Nevertheless, this solution still has a problem: it incurs a substantial cost for maintaining complete information about the load at every node in the system. In this paper, we propose a new framework, called Histogram-based Global Load

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

Balancing (HiGLOB) to facilitate global load balancing in structured P2P systems. Each node P in HiGLOB has two key components.

The first component is a histogram manager that maintains a histogram that reflects a global view of the distribution of the load in the system. The histogram stores statistical information that characterizes the average load of no overlapping groups of nodes in the P2P network. These nodes are connected to P through its neighbor nodes. The histogram information can be used for two purposes.

On one hand, it is used to determine if a node is normally loaded, overloaded, or under loaded. On the other hand, it is used to facilitate the discovery of a lightly loaded node or a heavily loaded node for the load-balancing process when it is needed. The second component of the system is a load-balancing manager that takes actions to redistribute the load whenever a node becomes overloaded or under loaded.

The load-balancing manager may redistribute the load both statically when a new node joins the system and dynamically when an existing node in the system becomes overloaded or under loaded. We exploit the routing metadata to partition the P2P network into no overlapping regions corresponding to the histogram buckets. We propose mechanisms to keep the cost of constructing and maintaining the histograms low.

To summarize, this paper makes the following contributions: It proposes a general framework that uses histograms to maintain a global view of the load distribution on structured P2P systems.

II. MODULES

There are the five modules present in the HI-GLOB framework.

1. Create Histogram
2. Send & get Request
3. Calculate load
4. Load Balance
5. Send & get Response

III. MODULE DESCRIPTION

1. CREATE HISTOGRAM

In this module, network is spited in to non-overlapping regions. Histogram Manager will maintain histogram for each peer. In that histogram, that statistical information of loads will be stored. Histogram of each peer contains the load values of neighbor peers.

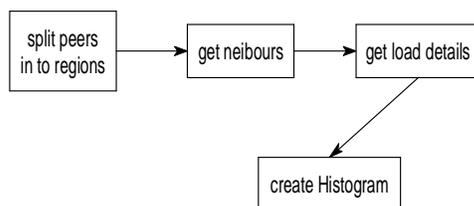


FIG.1 CREATE HISTOGRAM
2 SEND AND GET REQUEST

Using this module, each peer can request actions to another peer. The request may be like zip, unzip, compress, decompress and converting to document. The peer will get the request. This request will be considered as load. Before processing the request the load of the peer will be calculated.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

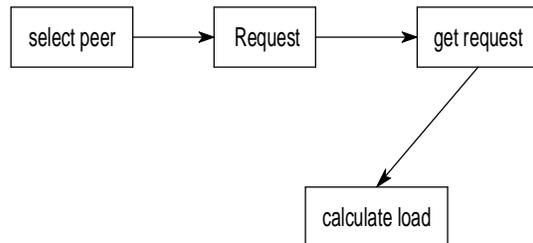


FIG.2 SEND AND GET REQUEST

3 CALCULATE LOAD

The load of the peer will be calculated before processing the request. The individual load of the peer will be calculated first. Then the average load the region will be calculated. If the individual load is greater than 2 times of average load then that peer will be considered as overloaded. Else the peer will be considered as normally loaded peer.

After calculated load the load factor will be calculated. Load factor is the ratio between the previous load and current load. If the load factor is greater than 1 the histogram will be updated. To reduce the histogram maintenance cost, the histogram no needs to be updated for small changes in loads. So if the load factor is less than 1 the histogram will not be updated.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

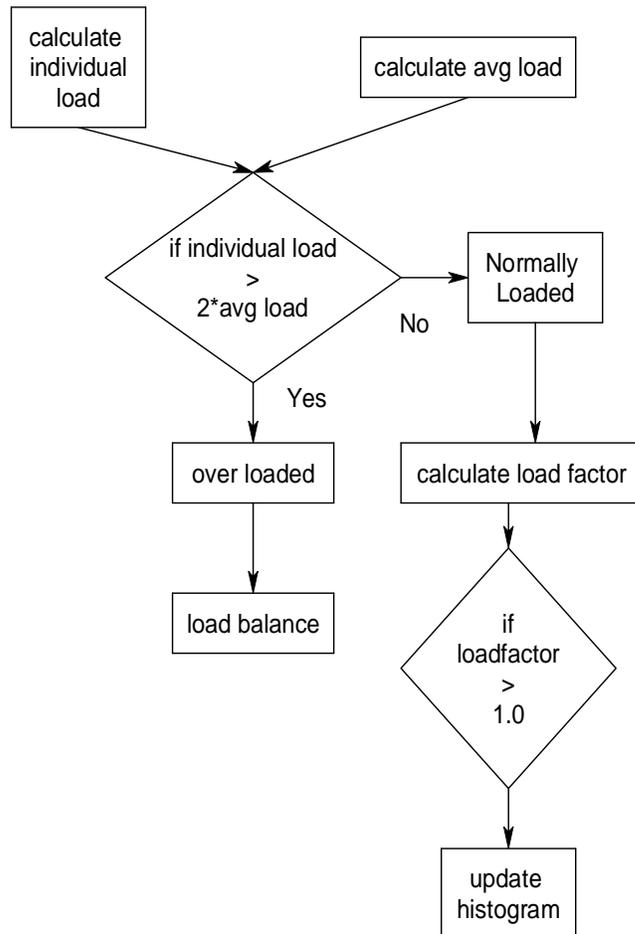


FIG 3-CALCULATING LOAD

4 LOAD BALANCE

If the peer is normally loaded, the load will be handled by the same peer. Else if it is overloaded the number of peers in the region will be counted. If there is only one peer the load will be shared that peer. If the number of peers are even the peers will be split as pair of peers, and the load is shared to the pair, which is having low load. If the number of peers is odd then the peers will be split as pair of peers and there will be one more odd peer. The load will be shared to either peer pair or to the odd peer, which is having fewer loads.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

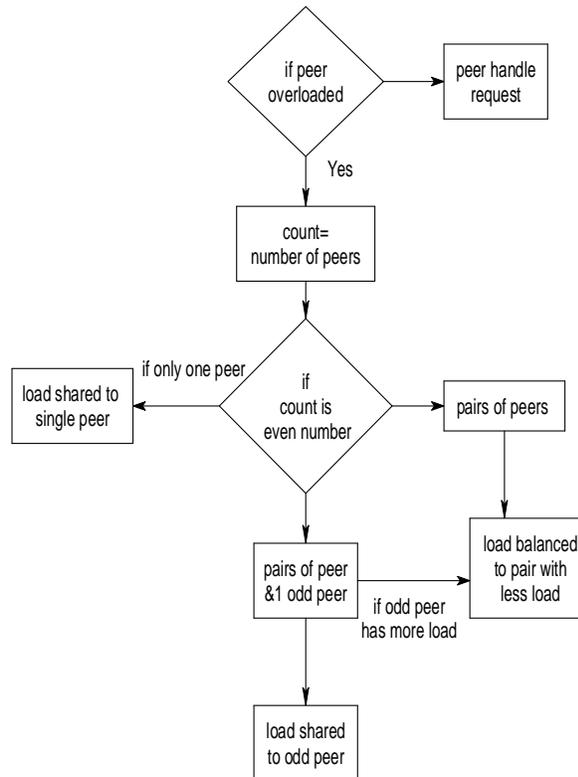
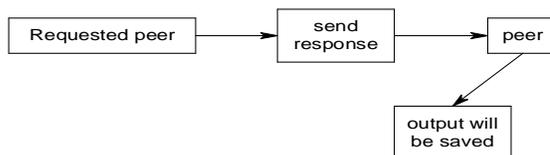


FIG 4 LOAD BALANCE

5 SEND AND GET RESPONSE

The response will be send to the peer. The response may be like decompressed file, compressed file, document, zipped folder or unzipped files. The response will be saved in the peer who sent request.



FIGS 5 SEND AND GET RESPONSE



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

IV. CONCLUSION

In this paper, we proposed a framework, HiGLOB, to enable global load balance for structured P2P systems. Each node in HiGLOB maintains the load information of nodes in the systems using histograms. This enables the system to have a global view of the load distribution and hence facilitates global load balancing. We partition the system into nonoverlapping groups of nodes and maintain the average load of them in the histogram at a node. We also proposed two techniques to reduce the overhead of maintaining and constructing histograms. Even though the proposal is a general framework, it is possible to deploy different kinds of P2P systems on it.

REFERENCES

- [1] D. Karger and M. Ruhl, "Simple Efficient Load Balancing Algorithms for Peer-to-Peer Systems," Proc. ACM Symp. Parallelism in Algorithms and Architectures (SPAA), 2004.
- [2] S3: Scalable, Shareable and Secure P2P Based Data Management System, <http://www.comp.nus.edu.sg/~s3p2p>, 2008.
- [3] Madhukar and C. Williamson, "A Longitudinal Study of P2P Traffic Classification," Proc. Int'l Symp. Modeling, Analysis, and Simulation of Computer and Telecomm. Systems (MASCOTS), 2006.
- [4] S. Saroiu, P.K. Gummadi, and S.D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," Proc. Multimedia Computing and Networking Conf. (MMCN), 2002.