Accomplishing of Load-Balancing In Peer-To-Peer Systems

Swathi Devathi[1*]., Sridhar Burada[2] and Vijaya Lakshmi, L[3]

1. Department of CSE, Malla Reddy College of Engineering, Misammaguda, Hyderabad, AP
2. Department of CSE, Medha College og Engineering, Bibi Nagar, Nalgonda, AP
3. Department of CSE, TRR College of Engineering, Inole, Patancheru, Hyderabad, AP

KEYWORDS

Peer-to-peer system; Virtual Server; Load-balancing; Distributed Hash Table

Abstract: *The systems of peer-to-peer have become known as an interesting solution intended for sharing and locating resources over the Internet. With the idea of virtual servers, peers who are participating in a, structured peer-to-peer network may possibly host various numbers of virtual servers, and by means of migrating virtual servers, peers can possibly stabilize their loads proportional to their capacity. Virtual server looks like a single peer towards the underlying distributed hash table, but each physical node can be accountable for more than one virtual server. Three schemes of simple load-balancing that are different primarily in the quantity of information used to choose how to rearrange load was introduced to attain the load balancing in the structured peer-to-peer systems.*

1. INTRODUCTION

Structured systems of peer-to-peer is mapped towards a unique identifier ID. Each object that enters the system has a connected load, movement cost, which is charged every time when the object between nodes was moved. An object's load may possibly or may not be connected to its movement cost. The essential approach towards the load balancing concern in structured peer to peer networks or distributed hash tables is consistent hashing [4]. If the hot peers turn out to be bottleneck, it leads to amplified user response time in addition to noteworthy performance degradation of the system and consequently the mechanism of load balancing is essential in such cases. With the idea of virtual servers, peers who are participating in a, structured peer-to-peer network may possibly host various numbers of virtual servers, and by means of migrating virtual servers, peers can possibly stabilize their loads proportional to their capacity [8]. The vulnerabilities of security are analyzed of the typical mechanism of distributed hash table load balancing. Various solutions have been introduced to tackle the problem of load balancing. On the other hand, these all believe that the system is static and for the most part of them assume that the IDs of both nodes in addition to items are consistently distributed. Virtual server looks like a single peer towards the underlying distributed hash table, but each physical node can be accountable for more than one virtual server [1]. For instance, in Chord, every virtual server is accountable for a contiguous region of the identifier space however a node can possess non-contiguous portions of the ring by means of containing numerous virtual servers. The important benefit of splitting load into virtual servers is that a virtual server can be moved from any node towards any other node within the system and this function looks like a leave followed by means of a join towards the underlying distributed hash table, and for this reason is supported by all distributed hash tables [11]. If each node has simply one virtual server, it can possibly transmit load towards nodes that are its neighbours within the ID space. Although splitting of load into virtual servers will enhance the length of path on the overlay, it was believed that the flexibility to progress load from whichever node towards any other node is critical to any scheme of load-balancing over distributed hash tables [3]. While algorithms of peer-to-peer are symmetric, specifically all peers play the similar role within

*** Mrs. SWATHI DEVATHI**
M.Tech Student
Department of Computer Science & Engg.
Malla Reddy College of Engineering,
Misammaguda, Hyderabad, A.P, India
Ph. No: 91- 9494598678
E- Mail: devathiswathi@gmail.com

the protocol; the systems of peer to peer can be extremely varied. A peer to peer system similar to Gnutella or else Kazaa may possibly consist of peers that range from old desktops following modem lines towards powerful servers associated to the Internet all the way through lines of high-bandwidth [14].

2. METHODOLOGY:

Three schemes of simple load-balancing that are different primarily in the quantity of information used to choose how to rearrange load was introduced. All these schemes attempt to stabilize the load by means of transferring virtual servers from nodes of heavily loaded towards lightly loaded nodes [9]. The important difference connecting these schemes is the amount of information necessary to make the decision of transfer. In the scheme of simplest, the transfer decision entails simply two nodes, while within the scheme of most complex, the transfer decision entails a set consisting of nodes of heavy and light. Assume that Nj denote the load of node j, where Nj corresponds to the sum of the loads of each virtual servers of node j [7]. Every node also has a target load Gi selected previously. A node is measured to be heavy if Nj > Gi and is light or else. In the Splitting of Virtual Servers if no virtual server in a heavy node can be moved within its entirety to an additional node, then a prospect is to split it into minor virtual servers and move a smaller virtual server towards a light node [2]. While this would get better the time taken to attain balance and probably reduce the total load moved, there is a threat of extremely fragmenting the identifier space. Enhancement in the number of virtual servers would involve improvement in the hop length of overlay and dimension of routing tables [12]. Consequently a scheme to periodically combine virtual servers would be essential to counteract increase in the number of virtual servers which is caused by means of splitting. The main difficulty for the scheme of one-to-one is the number of probes, which unconstructively impacts both the control traffic overhead in addition to convergence time [5]. Transfer of virtual server is commenced if one of the nodes is heavy as well as the other is light. This scheme is simple to put into practice in a distributed manner. Each light node can regularly choose a random ID and subsequently carry out a lookup operation to discover the node that is accountable for that ID. If that node is a heavy node, subsequently a transfer may possibly take place among the two nodes [10]. In this scheme only light nodes carry out probing; heavy nodes do not carry out any probing. There are three benefits of this choice of design. Initially, heavy nodes are reassured of the burden of undertaking the probing as well. Second, when the load of the system is extremely high and for the most part of the nodes is heavy, there is no risk of moreover overloading the network or else thrashing. Third, if the load of a node is related through the length of the ID space owned by means of that node, a random probe performed by a light node is additionally probable to discover a heavy node. The scheme of one-to-one achieves comparable results with respect to load moved and the excellence of balance achieved as the scheme of one-to-many. One-to-many permits a heavy node to regard as more than solitary light node at an instance. This scheme can be implemented by means of maintaining directories that accumulate load information about a set of light nodes within the system [13]. The similar distributed hash table system was used to accumulate these directories. A heavy node d examines the loads of a set of light nodes by means of contacting a random node of directory to which a random set of light nodes have sent their information of load. Some of the virtual servers of heavy nodes are subsequently moved to one or more of the lighter nodes scheduled within the directory. In the scheme of many-to-many each directory upholds the information of the load which is intended for a set of both nodes of light and heavy [6]. An algorithm run by means of each directory chooses the reassignment of virtual servers from heavy nodes recorded in that directory towards the nodes of light scheduled within that directory. This information of nodes' loads, which is additionally centralized than in the initial two systems, can be expected to make available an improved load balance. This scheme is a logical extension of the initial two systems. While in the initial scheme one heavy node was matched to a light node and in the second system one heavy node was matched to numerous light nodes, in this scheme many heavy nodes to numerous light nodes were matched.

3. RESULTS:

The scheme of one-to-one achieves comparable results with respect to load moved and the excellence of balance achieved as the scheme of one-to-many. The main difficulty for the scheme of one-to-one is the number of probes, which unconstructively impacts both the control traffic overhead in addition to convergence time.

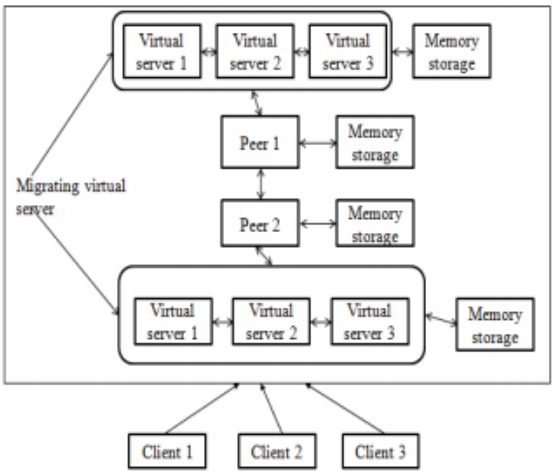

Fig.: Load Balance System Architecture

The total number of probes executed by heavy nodes before they entirely shed their excess load was plotted in the figure to enumerate the overhead. A probe is measured as useful if it outcomes in the transfer of a node of virtual node. It was shown that the scheme of one-to-one may possibly be satisfactory if loads stay on stable over long time scales, and if the overhead of control traffic does not have an effect on the system adversely. In the scheme of one-to-many and the many-to many, the fraction of load moved as well as the number of probes per node relies only on S/t, where S is the number of nodes within the system and t is the number of directories. This is for the reason that each directory encloses a random sample of size, S/t and the features of this sample do not rely on S for a large enough S. All three schemes perform extremely well in terms of scalability.

4. CONCLUSION:

While algorithms of peer-to-peer are symmetric, specifically all peers play the similar role within the protocol; the systems of peer to peer can be extremely varied. One-to-One Scheme is the initial scheme which is based on a mechanism of one-to-one rendezvous method, where two nodes are chosen at random. The scheme of one-to-one achieves comparable results with respect to load moved and the excellence of balance achieved as the scheme of one-to-many. One-to-many permits a heavy node to regard as more than solitary light node at an instance. In the scheme of many-to-many each directory upholds the information of the load which is intended for a set of both nodes of light and heavy.

5. REFERENCES:

[1] K. Hildrum and J. Kubiatowicz and S. Rao and B. Y. Zhao. “Distributed Object Location in a Dynamic Network”, Proc. ACM SPAA, 2002.

[2] J.R.Douceur &R.P.Wattenhofer. “Competitive Hill-Climbing Strategies for Replica Placement in a Distributed File System”, Lecture Notes in Computer Science, Vol. 2180, 2001.

[3] Frank Dabek, Frans Kaashoek, David Karger, Robert Morris, and Ion Stoica, “Wide-area Cooperative Storage with CFS,” in Proc. ACM SOSP, Banff, Canada, 2001.

[4] S. Ratnasamy and P. Francis and M. Handley and R. Karp and S. Shenker. “A Scalable Content-Addressable Network”, Proc. ACM SIGCOMM 2001.

[5] J. R. Douceur and R. P. Wattenhofer. “Optimizing File Availability in a Secure Serverless Distributed File System”, Proc. of 20th IEEE SRDS, 2001.

[6] Ananth Rao, Karthik Lakshminarayanan, Sonesh Surana, Richard Karp, and Ion Stoica, “Load Balancing in Structured P2P Systems,” in Proc. IPTPS, Feb. 2003.

[7] David Karger, Eric Lehman, Tom Leighton, Matthew Levine, Daniel Lewin, and Rina Panigrahy, “Consistent Hashing and andom Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web,” in Proc. ACM STOC, May 1997.

[8] I. Stoica and R. Morris and D. Karger M. F. Kaashoek H. Balakrishnan. “Chord: A scalable Peer-to-Peer Lookup Service for Internet Applications”, Proc. ACM SIGCOMM 2001.

[9] A. Rowstron and P. Druschel. “Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems”, Proc. IFIP/ACMInternational Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pp 329-350, Nov, 2001.

[10] F. Dabek and M. F. Kaashoek and D. R. Morris, I. Stoica. “Wide-area Cooperative Storage with CFS”, Proc. ACM SOSP 2001.

[11] M. Harren and J. M. Hellerstein and R. Huebsch and B. T. Loo, S. Shenker and I. Stoica. “Complex Queries in DHT-based Peerto-Peer Networks”, Proc. IPTPS 2002.

[12] P. Triantafillou and C. Xiruhaki and M. Koubarakis and N. Ntarmos. “Towards High Performance Peer-to-Peer Content and Resource Sharing Systems”, Proc. of CIDR, 2003.

[13] M. Andrews, M. X. Goemans, and L. Zhang, “Improved bounds for on-line load balancing,” in Proc. COCOON, 1996.

[14] G. Aggarwal, R. Mot “The Load Rebalancing Problem,” in Proc. ACM SPAA, 2003.