ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Balancing the Load to Reduce Network Traffic in Private Cloud

A.Shenbaga Bharatha Priya[1], J.Ganesh[2]

M-TECH (IT) Final Year, Department of IT, Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, Tamilnadu, India [1]

Asst. Prof, Department of IT, Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, Tamilnadu, India[2]

ABSTRACT - Infrastructure-As-A-Service (IAAS) provides an environmental setup under anyone type of cloud. In Distributed file system (DFS), nodes simultaneously serve computing and storage functions; that is parallel Data processing and storage in cloud. Here, file is considered as a data. That file is partitioned into a number of chunks allocated in distinct nodes so that MapReduce tasks can be performed in parallel over the nodes. Files and Nodes can be dynamically created, deleted, and appended. This results in load imbalance in a distributed file system; that is, the file chunks are not distributed as uniformly as possible among the nodes. Emerging distributed file systems in production systems strongly depend on a central node for chunk reallocation or Distributed node to maintain global knowledge of all chunks. This dependence is clearly inadequate in a large-scale, failure-prone environment because the central load balancer is put under considerable workload that is linearly scaled with the system size; it may thus become the performance bottleneck and the single point of failure and memory wastage in distributed nodes. In this paper, a fully distributed load rebalancing algorithm is presented to cope with the load imbalance problem. Our algorithm is compared against a centralized approach in a production system and a competing distributed solution presented in the paper.

KEYWORDS—Cloud Computing, Load Rebalancing, Distributed File System, Movement Cost, Network Traffic.

I. INTRODUCTION

Cloud Computing (or cloud for short) is a compelling technology. In clouds, clients can dynamically allocate their resources on-demand without sophisticated deployment and management of resources. Key enabling technologies for clouds include the Map Reduce programming paradigm [1], distributed file systems (e.g., [2], [3]),virtualization (e.g., [4], [5]), and so forth. These techniques emphasize scalability, so clouds (e.g., [6]) can be large in scale, and comprising entities can arbitrarily fail and join while maintaining system reliability. Distributed file systems are key building blocks for cloud computing applications based on the Map Reduce programming paradigm. In such file systems, nodes simultaneously serve computing and storage functions; a file is partitioned into a number of chunks allocated in distinct nodes so that MapReduce tasks can be performed in parallel over the nodes. In such an application, a cloud partitions the file into a large number of disjointed and fixed-size pieces (or file chunks) and assigns them to different cloud storage nodes (i.e., chunk servers). Each storage node (or node for short) then calculates the frequency of each unique word by scanning and parsing its local file chunks. In such a distributed file system, the load of a node is typically proportional to the number of file chunks the node possesses [22]. Because the files in a cloud can be arbitrarily created, deleted, and appended, and nodes can be upgraded, replaced and added in the file system [27], the file chunks are not distributed as uniformly as possible among the nodes. Load balance among storage nodes is a critical function in clouds. In a load-balanced cloud, the resources can be well utilized and provisioned, maximizing the performance of MapReduce-based applications. State-of-the-art distributed file systems (e.g., Google GFS[7],[8] and Hadoop HDFS [3]) in clouds rely on central nodes to manage the metadata information of the file systems and to balance the loads of storage nodes based on that metadata. The

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

centralized approach simplifies the design and implementation of a distributed file system. However, recent experience (e.g., [18]) concludes that when the number of storage nodes, the number of files and the number of accesses to files increase linearly, the central nodes (e.g., the master in Google GFS) become a performance bottleneck, as they are unable to accommodate a large number of file accesses due to clients and MapReduce applications. In this paper, we are interested in studying the load rebalancing problem in distributed file systems specialized for large-scale, dynamic and data-intensive clouds. (The terms "rebalance" and "balance" are interchangeable in this paper.) Such a large-scale cloud has hundreds or thousands of nodes (and may reach tens of thousands in the future).

Our objective is to allocate the chunks of files as uniformly as possible among the nodes such that no node manages an excessive number of chunks. [36]Additionally, we aim to reduce network traffic (or movement cost) caused by rebalancing the loads of nodes as much as possible to maximize the network bandwidth available to normal applications. Moreover, as failure is the norm, nodes are newly added to sustain the overall system performance [26],[33], resulting in the heterogeneity of nodes. Exploiting capable nodes to improve the system performance is, thus, demanded. Specifically, in this study, we suggest offloading the load rebalancing task to storage nodes by having the storage nodes balance their loads spontaneously. This eliminates the dependence on central nodes.

II. BACKGROUND

A. Virtualization:

Virtualization is the most profound change that PCs and servers have experienced, said Simon Crosby, chief technology officer for Citrix Systems' Data Center and Cloud Division [9]. "IT departments have long been at the mercy of the technical demands of legacy applications", explained Chris Van Dyke, [10] Microsoft's chief technology strategist for the oil and gas industry. "Now, rather than having to maintain older operating systems because of the needs of a legacy application, IT departments can take advantage of the performance and security gains in a new OS (in one virtual machine) while supporting legacy applications in another. Also, the process of deploying applications becomes simpler, because applications can be virtualized and deployed as a single virtual machine". [14]Virtualization technology lets a single PC or server simultaneously run multiple operating systems or multiple sessions of a single OS. This lets users put numerous applications even those that run on different operating systems on a single PC or server instead of having to host them on separate machines as in the past. The approach is thus becoming a common way for businesses and individuals to optimize their hardware usage by maximizing the number and kinds of jobs a single CPU can handle.[13].

B. Hyperviser:

IaaS software is low-level code that runs independent of an operating system called a hypervisor , and is responsible for taking inventory of hardware resources and allocating resources based on demand.[12]

C. I-A-A-S

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.[33] The consumer does not manage or control the underlying cloud physical infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components[32].

D. Private Cloud

The cloud infrastructure is operated solely for an organization.[34] It may be managed by the organization or a third party and may exist on premise or off premise.

E. Parellel Data Processing

Particular tasks of processing a job can be assigned to different types of virtual machines which are automatically instantiated and terminated during the job execution,parallel.[31].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

F. Distributed File System

Files are stored on different storage resources, but appear to users as they are put on a single location. A distributed file system should be transparent, fault-tolerant and scalable.[11]

III. LOAD BALANCING

Load balancing is the process of improving the performance of the system by shifting of workload among the processors. Workload of a machine means the total processing time it requires to execute all the tasks assigned to the machine [17]. Load balancing is done so that every virtual machine in the cloud system does the same amount of work throughout therefore increasing the throughput and minimizing the response time [15]. Load balancing is one of the important factors to heighten the working performance of the cloud service provider. Balancing the load of virtual machines uniformly means that anyone of the available machine is not idle or partially loaded while others are heavily loaded. One of the crucial issue of cloud computing is to divide the workload dynamically.[35] The benefits of distributing the workload includes increased resource utilization ratio which further leads to enhancing the overall performance thereby achieving maximum client satisfaction [19].

IV. EXISTING SYSTEM

State-of-the-art distributed file systems (e.g., Google GFS and Hadoop HDFS) in clouds rely on central nodes to manage the metadata information of the file systems and to balance the loads of storage nodes based on that metadata.[21] The centralized approach simplifies the design and implementation of a distributed file system.[16] However, recent experience concludes that when the number of storage nodes, the number of files and the number of accesses to files increase linearly, the central nodes (e.g., the master in Google GFS) become a performance bottleneck, as they are unable to accommodate a large number of file accesses due to clients and MapReduce applications.

V. PROPOSED SYSTEM

In this paper, we are interested in studying the load rebalancing problem in distributed file systems specialized for large-scale, dynamic and data-intensive clouds.[25] (The terms "rebalance" and "balance" are interchangeable in this paper.) Such a large-scale cloud has hundreds or thousands of nodes (and may reach tens of thousands in the future).Our objective is to allocate the chunks of files as uniformly as possible among the nodes such that no node manages an excessive number of chunks. Additionally, we aim to reduce network traffic (or movement cost) caused by rebalancing the loads of nodes as much as possible to maximize the network bandwidth available to normal applications. Moreover, as failure is the norm, nodes are newly added to sustain the overall system performance, resulting in the heterogeneity of nodes. Exploiting capable nodes to improve the system performance is, thus, demanded.[29].Our proposal not only takes advantage of physical network locality in the reallocation of file chunks to reduce the movement cost but also exploits capable nodes to improve the overall system performance.[30]

VI. MODULES

1. Module creation
2. DHT formulation
3. Load balancing algorithm

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

MODULES DESCRIPTION

1) Module Creation

Our objective is to allocate the modules of files as uniformly as possible among the nodes such that no node manages an excessive number of modules.[23] A file is partitioned into a number of modules allocated in different nodes so that Map Reduce Tasks can be performed in parallel over the nodes. Because the files in a cloud can be dynamically created, deleted, and appended, and nodes can be upgraded, replaced and added in the file system, the file modules are distributed as uniformly as possible among the nodes.

2) DHT Formulation

The module servers in our proposal are organized as a DHT network. Typical DHTs guarantee that if a node leaves, then its locally hosted modules are reliably migrated to its successor; if a node joins, then it allocates the modules whose IDs immediately precede the joining node from its successor to manage. DHT's, given that a unique handle (or identifier) is assigned to each file module[20]. The storage nodes are structured as a network based on distributed hash tables (DHTs), DHTs enable nodes to self-organize and repair while constantly offering lookup functionality in node dynamism, simplifying the system provision and management.

3) Load Balancing Algorithm

In our proposed algorithm, each module server node I first estimate whether it is under loaded (light) or overloaded (heavy) without global knowledge.[24] A node is light if the number of modules it hosts is smaller than the threshold. First of all we will find the lightest node to take the set of modules from heaviest node. So we can do the process without failure. Load balancing is a technique to distribute workload across many computers or network to achieve maximum resource utilization, maximize throughput, minimize response time, and avoid overload. The load equalization service is sometimes provided by dedicated software package or hardware, like a multilayer switch or name server.

VII. RESULTS

The expected result from this load rebalancing algorithm for Distributed File System using Eucalyptus in private cloud should be more efficiency and provide better performance. i.e., Maximize the utilization of nodes and minimize the ideal node, when compare to other existing load rebalancing algorithm.

In this Section, snapshoot of my project is briefly explained are as follows,

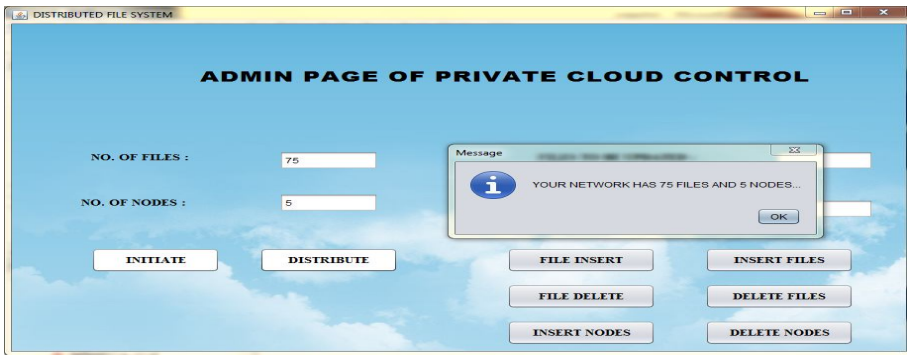

FIG 7.1 FRONT PAGE OF MY PROJECT

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

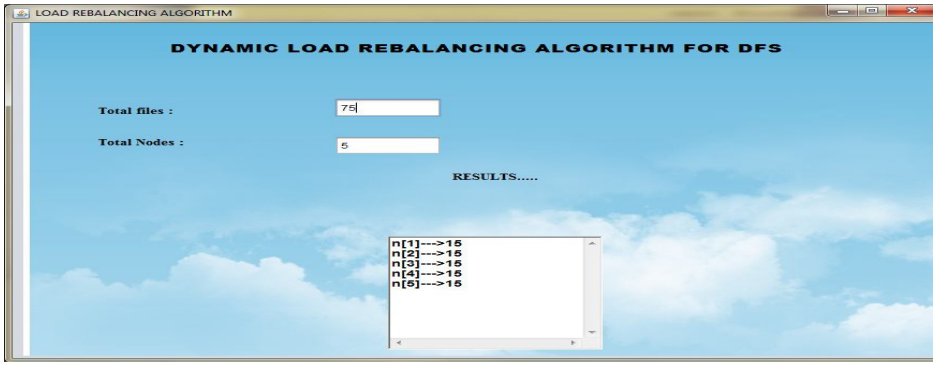

FIG 7.2 DYNAMIC LOAD REEBALANCING

FIG 7.3 FILE CHUNKS ALTERED

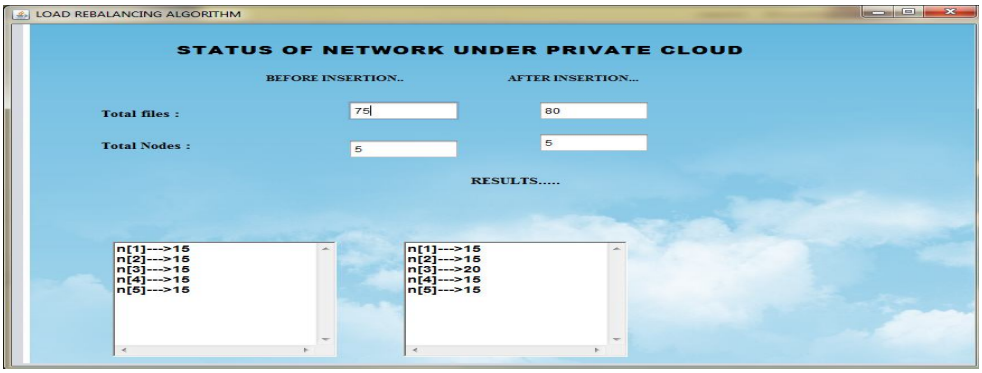

FIG 7.1 AFTER INSERTING FILE CHUNKS

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

FIG 7.1 AFTER LOAD REBALANCING

VIII. CONCLUSION

A novel load-balancing algorithm to deal with the load rebalancing problem in large-scale, dynamic, and distributed file systems in clouds has been presented in this paper.[28] Our proposal strives to balance the loads of nodes and reduce the demanded movement cost as much as possible, while taking advantage of physical network locality and node heterogeneity. In the absence of representative real workloads (i.e., the distributions of file chunks in a largescale storage system) in the public domain, we have investigated the performance of our proposal and compared it against competing algorithms through synthesized probabilistic distributions of file chunks. The synthesis workloads stress test the load-balancing algorithms by creating a few storage nodes that are heavily loaded. The computer simulation results are encouraging, indicating that our proposed algorithm performs very well.[37] Our proposal is comparable to the centralized algorithm in the Hadoop HDFS production system and dramatically outperforms the competing distributed algorithm in terms of load imbalance factor, movement cost, and algorithmic overhead. Particularly, our load-balancing algorithm exhibits a fast convergence rate. The efficiency and effectiveness of our design are further validated by analytical models and a real implementation with a small-scale cluster environment.

REFERENCES

[1] Hung-Chang Hsiao, Member, IEEE Computer Society, Hsueh-Yi Chung," Load Rebalancing for Distributed File Systems in Clouds",June 2013.
[2] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Proc. Sixth Symp. Operating System Design and Implementation (OSDI '04), pp. 137-150, Dec. 2004.
[3] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google File System," Proc. 19th ACM Symp. Operating Systems Principles (SOSP'03), pp. 29-43, Oct. 2003.
[4] Hadoop Distributed File System, http://hadoop.apache.org/hdfs/, 2012.
[5] VMware, http://www.vmware.com/, 2012.
[6] Xen, http://www.xen.org/, 2012.
[7] Apache Hadoop, http://hadoop.apache.org/, 2012.
[8] HDFS "Rebalancing Blocks," http:// developer.yahoo.com/hadoop/tutorial/module2.html#rebalancing,2012.
[9] K. McKusick and S. Quinlan, "GFS: Evolution on Fast-Forward," Comm. ACM, vol. 53, no. 3, pp. 42-49, Jan. 2010.
[10] Vishnu S. Pendyala , Synopsys Simon S.Y. Shim, San Jose State University, "The Web as the ubiquitous computer ,September 2009 , Web Technologies, journal of IEEE Computer Society"P 90-92.
[11] George Lawton., "Moving the OS to the Web,", March 2008, journal of ieee computer society, P 16-19.
[12] Benjamin Depardon,Cyril S_eguin,Ga□el Le Mahec "Analysis of Six Distributed File Systems",hal-00789086, version 1 - 15 Feb 2013.
[13] HDFS Federation, http://hadoop.apache.org/common/docs/ r0.23.0 / hadoop-yarn/hadoop-yarn-site/Federation.html, 2012.
[14] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek,F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-PeerLookup Protocol for Internet Applications," IEEE/ACM Trans.Networking, vol. 11, no. 1, pp. 17-21, Feb. 2003.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

[15] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems," Proc. IFIP/ACM Int'l Conf. Distributed Systems Platforms Heidelberg,pp. 161-172, Nov. 2001.
[16] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A.Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's Highly Available Key-Value Store,"Proc. 21st ACM Symp. Operating Systems Principles (SOSP '07), pp. 205-220, Oct. 2007.
[17] A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica,"Load Balancing in Structured P2P Systems," Proc. Second Int'lWorkshop Peer-to-Peer Systems (IPTPS '02), pp. 68-79, Feb. 2003.
[18] D. Karger and M. Ruhl, "Simple Efficient Load Balancing Algorithms for Peer-to-Peer Systems," Proc. 16th ACM Symp.Parallel Algorithms and Architectures (SPAA '04), pp. 36-43, June 2004.
[19] P. Ganesan, M. Bawa, and H. Garcia-Molina, "Online Balancing of Range-Partitioned Data with Applications to Peer-to-Peer Systems,"Proc. 13th Int'l Conf. Very Large Data Bases (VLDB '04),pp. 444-455, Sept. 2004.
[20] J.W. Byers, J. Considine, and M. Mitzenmacher, "Simple Load Balancing for Distributed Hash Tables," Proc. First Int'l Workshop Peer-to-Peer Systems (IPTPS '03), pp. 80-87, Feb. 2003.
[21] G.S. Manku, "Balanced Binary Trees for ID Management and Load Balance in Distributed Hash Tables," Proc. 23rd ACM Symp.Principles Distributed Computing (PODC '04), pp. 197-205, July 2004.
[22] A. Bharambe, M. Agrawal, and S. Seshan, "Mercury: Supporting Scalable Multi-Attribute Range Queries," Proc. ACM SIGCOMM '04, pp. 353-366, Aug. 2004.
[23] Y. Zhu and Y. Hu, "Efficient, Proximity-Aware Load Balancing for DHT-Based P2P Systems," IEEE Trans. Parallel and Distributed Systems, vol. 16, no. 4, pp. 349-361, Apr. 2005.
[24] Q.H. Vu, B.C. Ooi, M. Rinard, and K.-L. Tan, "Histogram-Based Global Load Balancing in Structured Peer-to-Peer Systems," IEEE Trans. Knowledge Data Eng., vol. 21, no. 4, pp. 595-608, Apr. 2009
[25] Gaochao Xu, Junjie Pang, and Xiaodong Fu"A Load Balancing Model Based on Cloud Partitioning for the Public Cloud", IEEE TRANSACTIONS ON CLOUD COMPUTING YEAR 2013.
[26] H. Shen and C.-Z. Xu, "Locality-Aware and Churn-Resilient Load Balancing Algorithms in Structured P2P Networks," IEEE Trans.Parallel and Distributed Systems, vol. 18, no. 6, pp. 849-862, June 2007.
[27] Q.H. Vu, B.C. Ooi, M. Rinard, and K.-L. Tan, "Histogram-Based Global Load Balancing in Structured Peer-to-Peer Systems," IEEE Trans. Knowledge Data Eng., vol. 21, no. 4, pp. 595-608, Apr. 2009.
[28] H.-C. Hsiao, H. Liao, S.-S. Chen, and K.-C. Huang, "Load Balancewith Imperfect Information in Structured Peer-to-Peer Systems," IEEE Trans. Parallel Distributed Systems, vol. 22, no. 4, pp. 634-649,Apr. 2011.
[29] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Co., 1979.
[30] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-Based Aggregation in Large Dynamic Networks," ACM Trans. Computer Systems, vol. 23, no. 3, pp. 219-252, Aug. 2005.
[31] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M.V. Steen, "Gossip-Based Peer Sampling," ACM Trans. Computer Systems, vol. 25, no. 3, Aug. 2007.
[32] H. Sagan, Space-Filling Curves, first ed. Springer, 1994.
[33] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube: A High Performance, Server-Centric Network Architecture for Modular Data Centers," Proc. ACM SIGCOMM '09, pp. 63-74, Aug. 2009.
[34] H. Abu-Libdeh, P. Costa, A. Rowstron, G. O'Shea, and A.Donnelly, "Symbiotic Routing in Future Data Centers," Proc.ACM SIGCOMM '10, pp. 51-62, Aug. 2010.
[35] S. Surana, B. Godfrey, K. Lakshminarayanan, R. Karp, and I. Stoica, "Load Balancing in Dynamic Structured P2P Systems,"Performance Evaluation, vol. 63, no. 6, pp. 217-240, Mar. 2006.
[36] S. Iyer, A. Rowstron, and P. Druschel, "Squirrel: A Decentralized Peer-to-Peer Web Cache," Proc. 21st Ann. Symp. Principles of Distributed Computing (PODC '02), pp. 213-222, July 2002.
[37] I. Raicu, I.T. Foster, and P. Beckman, "Making a Case for Distributed File Systems at Exascale," Proc. Third Int'l Workshop Large-Scale System and Application Performance (LSAP '11), pp. 11- 18, June 2011.