Journal of Information Systems and Communication
ISSN: 0976-8742, E-ISSN: 0976-8750, Volume 3, Issue 1, 2012, pp- 87-91.
Available online at http://www.bioinfo.in/contents.php?id=45

EXISTING LOAD BALANCING TECHNIQUES IN CLOUD COMPUTING: A SYSTEMATIC REVIEW

NIDHI JAIN KANSAL* AND INDERVEER CHANA

Department of Computer Science and Engineering, Thapar University, Patiala, India - 147001
*Corresponding Author: Email- nidhi.kavya@gmail.com

Received: January 12, 2012; Accepted: February 15, 2012

Abstract- In cloud computing, load balancing is required to distribute the dynamic local workload evenly across all the nodes. It helps to achieve a high user satisfaction and resource utilization ratio by ensuring an efficient and fair allocation of every computing resource. Proper load balancing aids in minimizing resource consumption, implementing fail-over, enabling scalability, avoiding bottlenecks and over-provisioning etc. In this paper, a systematic review of existing load balancing techniques is presented. Out of 3,494 papers analyzed, 15 papers are identified reporting on 17 load balancing techniques in cloud computing. This study concludes that all the existing techniques mainly focus on reducing associated overhead, service response time and improving performance etc. Various parameters are also identified, and these are used to compare the existing techniques.
Keywords- Cloud Computing, Load Balancing, Green Computing

Citation: Nidhi Jain Kansal and Inderveer Chana (2012) Existing Load Balancing Techniques In Cloud Computing: A Systematic Review. Journal of Information Systems and Communication. ISSN: 0976-8742, E-ISSN: 0976-8750, Volume 3, Issue 1, pp- 87-91.

Introduction

Cloud computing, a framework for enabling convenient, and on-demand network access to a shared pool of computing resources [17], is emerging as a new paradigm of large-scale distributed computing [16]. It has widely been adopted by the industry, though there are many existing issues like Load Balancing, Virtual Machine Migration, Server Consolidation, Energy Management, etc. that are not fully addressed [18]. Central to these issues is the issue of load balancing that is a mechanism to distribute the dynamic workload evenly to all the nodes in the whole cloud to achieve a high user satisfaction and resource utilization ratio [10]. It helps in preventing bottlenecks of the system which may occur due to load imbalance. When one or more components of any service fail, load balancing facilitates continuation of the service by implementing fair-over, i.e. it helps in provisioning and de-provisioning of instances of applications without fail. It also ensures that every computing resource is distributed efficiently and fairly [18] [23]. Consumption of resources and conservation of energy is not always a prime focus of discussion in cloud computing. However, resource consumption can be kept to a minimum with proper load balancing which not only helps in reducing costs but making enterprises greener [19] [22]. Scalability, one of the very important features of cloud computing, is also enabled by load balancing. Hence, improving resource utility and the performance of a distributed system in such a way will reduce the energy consumption and carbon footprints to achieve Green computing [20] [21].

The objective and motivation of this survey is to give a systematic review of existing load balancing techniques in cloud computing and encourage the amateur researcher in this field, so that they can contribute in developing more efficient load balancing algorithm. This will benefit interested researchers to carry out further work in this research area. The rest of the paper is organized as follows: Section II focuses on the research method used in this study. Section III discusses about the existing load balancing techniques in cloud computing. Section IV identifies the metrics and does the comparison of techniques based on them. Section V lists the different publication fora in which the articles have been published and shows the analysis done on the existing load balancing techniques. Finally, section V concludes the paper.

Research Method

This review aims at summarizing the current state of the art of existing load balancing techniques in cloud computing.

Sources of Information

The search was widely conducted in the following electronic sources to gain a broad perspective

- ACM Digital Library (portal.acm.org)
- IEEE eXplore (ieeexplore.ieee.org)
- ScienceDirect (www.sciencedirect.com)
- Springer LNCS (www.springer.com/lncs)

These sources cover the most relevant journals, conferences and workshop proceedings. The searches in the selected sources resulted in overlap among the papers, where the duplicates were excluded primarily by manual filtering.

Search criteria

The initial search criteria included the titles (load balancing in cloud computing), (load balancing techniques in cloud computing), (load balancing in clouds) and (load balancing in datacenters). The start year was set to 2006, and the end year was 2011. Only papers written in English were included. The initial search located 3,494 potentially relevant papers.

Study Selection

A selection process was used which included three stages as shown in "Fig. (1)". Based on titles, the papers

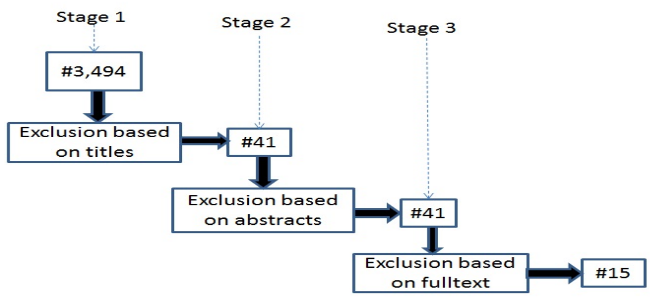

Fig. 1- Study Selection Procedure

which were duplicate and irrelevant, were excluded in the first stage. After the first stage, 41 papers remained. In the second stage, information in abstracts was analyzed and all the 41 papers seemed to be relevant. In the third stage a full text analysis was performed. After abstract and full text analysis, 15 papers were finally selected that reported on 17 load balancing techniques which could be implemented in cloud computing environment.

Existing Load Balancing Techniques In Clouds

Following load balancing techniques are currently prevalent in clouds

- **VectorDot-** A. Singh et al. [1] proposed a novel load balancing algorithm called VectorDot. It handles the hierarchical complexity of the data-center and multidimensionality of resource loads across servers, network switches, and storage in an agile data center that has integrated server and storage virtualization technologies. VectorDot uses dot product to distinguish nodes based on the item requirements and helps in removing overloads on servers, switches and storage nodes.
- **CARTON-** R. Stanojevic et al. [2] proposed a mechanism CARTON for cloud control that unifies the use of LB and DRL. LB (Load Balancing) is used to equally distribute the jobs to different servers so that the associated costs can be minimized and DRL (Distributed Rate Limiting) is used to make sure that the resources are distributed in a way to keep a fair resource allocation. DRL also adapts to server capacities for the dynamic workloads so that performance levels at all servers are equal. With very low computation and communication overhead, this algorithm is simple and easy to implement.
- **Compare and Balance-** Y. Zhao et al. [3] addressed the problem of intra-cloud load balancing amongst physical hosts by adaptive live migration of virtual machines. A load balancing model is designed and implemented to reduce virtual machines' migration time by shared storage, to balance load amongst servers according to their processor or IO usage, etc. and to keep virtual machines' zero-downtime in the process. A distributed load balancing algorithm COMPARE AND BALANCE is also proposed that is based on sampling and reaches equilibrium very fast. This algorithm assures that the migration of VMs is always from high-cost physical hosts to low-cost host but assumes that each physical host has enough memory which is a weak assumption.
- **Event-driven-** V. Nae et al. [4] presented an event-driven load balancing algorithm for real-time Massively Multiplayer Online Games (MMOG). This algorithm after receiving capacity events as input, analyzes its components in context of the resources and the global state of the game session, thereby generating the game session load balancing actions. It is capable of scaling up and down a game session on multiple resources according to the variable user load but has occasional QoS breaches.
- **Scheduling strategy on LB of VM resources -** J. Hu et al. [5] proposed a scheduling strategy on load balancing of VM resources that uses historical data and current state of the system. This strategy achieves the best load balancing and reduced dynamic migration by using a genetic algorithm. It helps in resolving the issue of load imbalance and high cost of migration thus achieving better resource utilization.
- **CLBVM-** A. Bhadani et al. [6] proposed a Central Load Balancing Policy for Virtual Machines (CLBVM) that balances the load evenly in a distributed virtual machine/cloud computing environment. This policy improves the overall performance of the system but does not consider the systems that are fault-tolerant.
- **LBVS-** H. Liu et al. [7] proposed a load balancing virtual storage strategy (LBVS) that provides a large scale net data storage model and Storage as a Service model based on Cloud Storage. Storage virtualization is achieved using an architecture that is three-layered and load balancing is achieved using two load balancing modules. It helps in improving the efficiency of concurrent access by using replica balancing further reducing the response time and enhancing the capacity of disaster recovery. This strategy also helps in improving the use rate of storage resource, flexibility and robustness of the system.

- **Task Scheduling based on LB-** Y. Fang et al. [8] discussed a two-level task scheduling mechanism based on load balancing to meet dynamic requirements of users and obtain high resource utilization. It achieves load balancing by first mapping tasks to virtual machines and then virtual machines to host resources thereby improving the task response time, resource utilization and overall performance of the cloud computing environment.
- **Honeybee Foraging Behavior-** M. Randles et al. [9] investigated a decentralized honeybee-based load balancing technique that is a nature-inspired algorithm for self-organization. It achieves global load balancing through local server actions. Performance of the system is enhanced with increased system diversity but throughput is not increased with an increase in system size. It is best suited for the conditions where the diverse population of service types is required.
- **Biased Random Sampling-** M. Randles et al. [9] investigated a distributed and scalable load balancing approach that uses random sampling of the system domain to achieve self-organization thus balancing the load across all nodes of the system. The performance of the system is improved with high and similar population of resources thus resulting in an increased throughput by effectively utilizing the increased system resources. It is degraded with an increase in population diversity.
- **Active Clustering-** M. Randles et al. [9] investigated a self-aggregation load balancing technique that is a self-aggregation algorithm to optimize job assignments by connecting similar services using local re-wiring. The performance of the system is enhanced with high resources thereby increasing the throughput by using these resources effectively. It is degraded with an increase in system diversity.
- **ACCLB-** Z. Zhang et al. [10] proposed a load balancing mechanism based on ant colony and complex network theory (ACCLB) in an open cloud computing federation. It uses small-world and scale-free characteristics of a complex network to achieve better load balancing. This technique overcomes heterogeneity, is adaptive to dynamic environments, is excellent in fault tolerance and has good scalability hence helps in improving the performance of the system.
- **(OLB + LBMM)-** S.-C. Wang et al. [11] proposed a two-phase scheduling algorithm that combines OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms to utilize better executing efficiency and maintain the load balancing of the system. OLB scheduling algorithm, keeps every node in working state to achieve the goal of load balance and LBMM scheduling algorithm is utilized to minimize the execution time of each task on the node thereby minimizing the overall completion time. This combined approach hence helps in an efficient utilization of resources and enhances the work efficiency.
- **Decentralized content aware-** H. Mehta et al. [12] proposed a new content aware load balancing policy named as workload and client aware policy (WCAP). It uses a parameter named as USP to specify the unique and special property of the requests as well as computing nodes. USP helps the scheduler to decide the best suitable node for processing the requests. This strategy is implemented in a decentralized manner with low overhead. By using the content information to narrow down the search, it improves the searching performance overall performance of the system. It also helps in reducing the idle time of the computing nodes hence improving their utilization.
- **Server-based LB for Internet distributed services-** A. M. Nakai et al. [13] proposed a new server-based load balancing policy for web servers which are distributed all over the world. It helps in reducing the service response times by using a protocol that limits the redirection of requests to the closest remote servers without overloading them. A middleware is described to implement this protocol. It also uses a heuristic to help web servers to endure overloads.
- **Join-Idle-Queue-** Y. Lua et al. [14] proposed a Join-Idle-Queue load balancing algorithm for dynamically scalable web services. This algorithm provides large-scale load balancing with distributed dispatchers by, first load balancing idle processors across dispatchers for the availability of idle processors at each dispatcher and then, assigning jobs to processors to reduce average queue length at each processor. By removing the load balancing work from the critical path of request processing, it effectively reduces the system load, incurs no communication overhead at job arrivals and does not increase actual response time.
- **Lock-free multiprocessing solution for LB-** X. Liu et al. [15] proposed a lock-free multiprocessing load balancing solution that avoids the use of shared memory in contrast to other multiprocessing load balancing solutions which use shared memory and lock to maintain a user session. It is achieved by modifying Linux kernel. This solution helps in improving the overall performance of load balancer in a multi-core environment by running multiple load-balancing processes in one load balancer.

Table I shows the existing load balancing techniques in cloud computing. This review identifies the techniques, their environment and description according to what has been stated in each of the selected papers.

Metrics For Load Balancing In Clouds

Various metrics considered in existing load balancing techniques in cloud computing are discussed below

- **Throughput** is used to calculate the no. of tasks whose execution has been completed. It should be high to improve the performance of the system.
- **Overhead Associated** determines the amount of overhead involved while implementing a load-balancing algorithm. It is composed of overhead due to movement of tasks, inter-processor and inter-process communication. This should be minimized so that a load balancing technique can work efficiently.
- **Fault Tolerance** is the ability of an algorithm to perform uniform load balancing in spite of arbitrary node or link failure. The load balancing should be a good fault-tolerant technique.
- **Migration time** is the time to migrate the jobs or resources from one node to other. It should be minimized in order to enhance the performance of the system.

- **Response Time** is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.
- **Resource Utilization** is used to check the utilization of resources. It should be optimized for an efficient load balancing.
- **Scalability** is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.
- **Performance** is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays.

Based on these metrics, the existing load balancing techniques have been compared in Table 2.

Table 2- Comparison Of Existing Load Balancing Techniques Based On Various Metrics

Metrics\ Techniques	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17
Throughput	×	×	×	×	×	✓	×	×	✓	✓	✓	×	×	×	×	×	✓
Overhead	×	✓	✓	×	✓	×	×	×	×	×	×	×	×	✓	×	✓	×
Fault tolerance	×	×	×	×	×	×	✓	×	×	×	×	✓	×	×	×	×	×
Migration time	×	×	✓	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Response Time	×	×	×	×	×	✓	✓	✓	×	×	×	×	×	✓	✓	✓	×
Resource Utilization	✓	✓	✓	✓	✓	✓	×	✓	×	×	×	✓	✓	✓	×	×	×
Scalability	×	×	×	✓	×	×	✓	×	✓	✓	✓	✓	×	✓	✓	×	×
Performance	×	✓	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Results and Discussions

The goal of this study was to determine whether the literature on load balancing techniques in cloud computing provides a uniform and rigorous base. The papers were initially obtained in a broad search in four databases covering relevant journals, conference and workshop proceedings. Then an extensive systematic selection process was carried out to identify papers describing load balancing techniques in cloud computing. The results presented here thus give a good picture of the existing load balancing techniques in cloud computing. Of 3,494 papers analyzed in the systematic review, 15 papers were identified on load balancing techniques which could be implemented in cloud computing. The papers report on 17 different techniques for load balancing in cloud computing.

Table 3 lists the different publication fora in which the articles have been published and Table IV shows the analysis done on the existing load balancing techniques.

Conclusion

Load balancing is one of the main challenges in cloud computing. It is required to distribute the dynamic local workload evenly across all the nodes to achieve a high user satisfaction and resource utilization ratio by making sure that every computing resource is distributed efficiently and fairly. With proper load balancing, resource consumption can be kept to a minimum which will further reduce energy consumption and carbon emission rate which is a dire need of cloud computing. Existing load balancing techniques that have been discussed mainly focus on reducing associated overhead, service response time and improving performance etc. but none of the techniques has considered the energy consumption and carbon emission factors.

Table 3- Number Of Papers In Different Publication Fora

Publication fora	Type	#	%
International Conference on Advanced Information Networking and Applications Workshops (WAINA)	Conference	1	6.67
International Conference on Industrial Mechatronics and Automation(ICIMA)	Conference	1	6.67
International Workshop on Intelligent Systems and Applications (ISA)	Conference	1	6.67
International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)	Conference	1	6.67
International Joint Conference on INC, IMS and IDC (NCM)	Conference	1	6.67
IEEE/ACM International Conference on Grid Computing (GRID)	Conference	1	6.67
International Conference on Service Sciences (ICSS)	Conference	1	6.67
Latin-American Symposium on Dependable Computing (LADC)	Conference	1	6.67
International Conference on Communications (ICC)	Conference	1	6.67
International Conference on Computer Science and Information Technology (ICCSIT)	Conference	1	6.67
Annual ACM Bangalore Conference (Compute)	Conference	1	6.67
International Conference & Workshop on Emerging Trends in Technology (ICWET)	Conference	1	6.67
ACM/IEEE conference on Supercomputing (SC)	Conference	1	6.67
An International Journal of Performance Evaluation	Journal	1	6.67
Web Information Systems and Mining, volume 6318	Lecture Notes in Comp. Sc.	1	6.67
Total		15	100

Table 4- Analysis Of Existing Load Balancing Techniques

Techniques	Findings
T1 [1]	1. Handles hierarchical and multidimensional resource constraints 2. Removes overloads on server, switch and storage
T2 [2]	1. Simple 2. Easy to implement 3. Very low computation and communication overhead
T3 [3]	1. Balances load amongst servers 2. Reaches equilibrium fast 3. Assures migration of VMs from high-cost physical hosts to low-cost host 4. Assumption of having enough memory with each physical host
T4 [4]	1. Capable of scaling up and down a game session on multiple resources according to the variable user load 2. Occasional QoS breaches
T5 [5]	1. Solves the problems of load imbalance and high migration cost
T6 [6]	1. Balances the load evenly to improve overall performance 2. Does not consider fault tolerance
T7 [7]	1. Enhances flexibility and robustness 2. Provides large scale net data storage and storage as a service
T8 [8]	1. Improves task response time 2. Improves resource utilization
T9 [9]	1. Performs well as system diversity increases 2. Does not increase throughput as system size increases
T10 [9]	1. Performs better with high and similar population of resources 2. Degrades as population diversity increases
T11 [9]	1. Performs better with high resources 2. Utilizes the increased system resources to increase throughput 3. Degrades as system diversity increases
T12 [10]	1. Overcomes heterogeneity 2. Adaptive to dynamic environments 3. Excellent in fault tolerance 4. Good scalability
T13 [11]	1. Efficient utilization of resources 2. Enhances work efficiency
T14 [12]	1. Improves the searching performance hence increasing overall performance 2. Reduces idle time of the nodes
T15 [13]	1. Reduces service response times by redirecting requests to the closest server without overloading them
T16 [14]	1. Effectively reduces the system load 2. Incurs no communication overhead at job arrivals 3. Does not increase actual response times
T17 [15]	1. Improves overall performance of load balancer

Therefore, there is a need to develop an energy-efficient load balancing technique that can improve the performance of cloud computing by balancing the workload across all the nodes in the cloud along with maximum resource utilization, in turn reducing energy consumption and carbon emission to an extent which will help to achieve Green computing.

References

[1] Singh A., Korupolu M. and Mohapatra D. (2008) *ACM/IEEE conference on Supercomputing*.

[2] Stanojevic R. and Shorten R. (2009) *IEEE ICC,* 1-6.

[3] Zhao Y. and Huang W. (2009) 5*th International Joint Conference on INC, IMS and IDC,* 170-175.

[4] Nae V., Prodan R. and Fahringer T. (2010) 11*th IEEE/ACM International Conference on Grid Computing (Grid),* 9-17.

[5] Hu J., Gu J., Sun G. and Zhao T. (2010) 3*rd International Symposium on Parallel Architectures, Algorithms and Programming*, 89-96.

[6] Bhadani A. and Chaudhary S. (2010) 3*rd Annual ACM Bangalore Conference*.

[7] Liu H., Liu S., Meng X., Yang C. and Zhang Y. (2010) *International Conference on Service Sciences (ICSS),* 257-262.

[8] Fang Y., Wang F. and Ge J. (2010) *Lecture Notes in Computer Science*, 6318, 271-277.

[9] Randles M., Lamb D. and Taleb-Bendiab A. (2010) 24*th International Conference on Advanced Information Networking and Applications Workshops*, 551-556.

[10] Zhang Z. and Zhang X. (2010) 2*nd International Conference on Industrial Mechatronics and Automation*, 240-243.

[11] Wang S., Yan K., Liao W. and Wang S. (2010) 3*rd International Conference on Computer Science and Information Technology*, 108-113.

[12] Mehta H., Kanungo P. and Chandwani M. (2011) *International Conference Workshop on Emerging Trends in Technology*, 370-375.

[13] Nakai A.M., Madeira E. and Buzato L.E. (2011) 5*th Latin-American Symposium on Dependable Computing*, 156-165.

[14] Lua Y., Xiea Q., Kliotb G., Gellerb A., Larusb J. R. and Greenber A. (2011) *Int. Journal on Performance evaluation.*

[15] Liu Xi., Pan Lei., Wang Chong-Jun. and Xie Jun-Yuan. (2011) 3*rd International Workshop on Intelligent Systems and Applications*, 1-4.

[16] Armbrust M., Fox A., Griffith R., Joseph A. D., Katz R., Konwinski A., Lee G., Patterson D., Rabkin A., Stoica I. and Zaharia M. (2009) *Above the Clouds: A Berkeley View of Cloud Computing, EECS Department, University of California*, 1-23.

[17] Pallis G. (2010) *IEEE J. of Internet Computing*, 14(5), 70-73.

[18] Rimal B.P., Choi E. and Lumb I. (2009) 5*th International Joint Conference on INC, IMS and IDC,* 44-51.

[19] Mata-Toledo R. and Gupta P. (2010) *Journal of Technology Research,* 2(1), 1-8.

[20] Kabiraj S., Topkar V. and Walke R.C. (2010) *International Journal of Managing Information Technology*, 2(3), 22-31.

[21] Baliga J., Ayre R.W.A., Hinton K. and Tucker R.S. (2011) *IEEE*, 99(1), 149-167.

[22] Alakeel A. M. (2010) *International Journal of Computer Science and Network Security,* 10(6), 153-160.

[23] Rimal B.P., Choi E. and Lumb I. (2010) *Computer Communications and Networks,* 21-46.

Table 1- Existing Load Balancing Techniques

S. No.	Techniques	Author/ Year	Environment	Description
T1[1]	VectorDot	A. Singh et al. / 2008	Datacenters with integrated server and storage virtualization	1. Uses dot product to distinguish node based on the item requirement
T2[2]	Carton	R. Stanojevic et al. / 2009	Unifying framework for cloud control	1. Uses Load Balancing to minimize the associated cost and uses Distributed Rate Limiting for fair allocation of resources
T3[3]	COMPARE AND BALANCE	Y. Zhao et al. / 2009	Intra-Cloud	1. Based on sampling 2. Uses adaptive live migration of virtual machines
T4[4]	Event-driven	V. Nae et al. / 2010	Massively Multiplayer Online Games	1. Uses complete capacity event as input, analyzes its components and generates the game session load balancing actions
T5[5]	Scheduling strategy on LB of VM resources	J. Hu et al. / 2010	Cloud Computing	1. Uses Genetic algorithm, historical data and current state of system to achieve best load balancing and to reduce dynamic migration
T6[6]	CLBVM	A. Bhadani et al. / 2010	Cloud Computing	1. Uses global state information to make load balancing decisions
T7[7]	LBVS	H. Liu et al. / 2010	Cloud Storage	1. Uses Fair-Share Replication strategy to achieve Replica Load balancing module which in turn controls the access load balancing 2. Uses writing balancing algorithm to control data writing load balancing
T8[8]	Task Scheduling based on LB	Y. Fang et al. / 2010	Cloud Computing	1. First maps tasks to virtual machines and then virtual machines to host resource
T9[9]	Honeybee Foraging Behavior	M. Randles et al. / 2010	Large scale Cloud Systems	1. Achieves global load balancing through local server action
T10[9]	Biased Random Sampling	M. Randles et al. / 2010	Large scale Cloud systems	1. Achieves load balancing across all system nodes using random sampling of the system domain
T11[9]	Active Clustering	M. Randles et al. / 2010	Large scale Cloud systems	1. Optimizes job assignment by connecting similar services by local re-wiring
T12[10]	ACCLB	Z. Zhang et al. / 2010	Open Cloud Computing Federation	1. Uses small-world and scale-free characteristics of complex network to achieve better load balancing
T13[11]	(OLB + LBMM)	S.-C. Wang et al. / 2010	Three-level Cloud Computing Network	1. Uses OLB (Opportunistic Load Balancing) to keep each node busy and uses LBMM (Load Balance Min-Min) to achieve the minimum execution time of each task
T14[12]	Decentralized content aware	H. Mehta et al. / 2011	Distributed computing	1. Uses a unique and special property(USP) of requests and computing nodes to help scheduler to decide the best node for processing the requests 2. Uses the content information to narrow down the search
T15[13]	Server-based LB for Internet distributed services	A. M. Nakai et al. / 2011	Distributed web servers	1. Uses a protocol to limit redirection rates to avoid remote servers overloading 2. Uses a middleware to support this protocol 3. Uses a heuristic to tolerate abrupt load changes
T16[14]	Join-Idle-Queue	Y. Lua et al. / 2011	Cloud data centers	1. First assigns idle processors to dispatchers for the availability of the idle processors at each dispatcher 2. Then assigns jobs to processors to reduce average queue length of jobs at each processor
T17[15]	Lock-free multiprocessing solution for LB	X. Liu et al. /2011	Multi-core	1. Runs multiple load-balancing processes in one load balancer