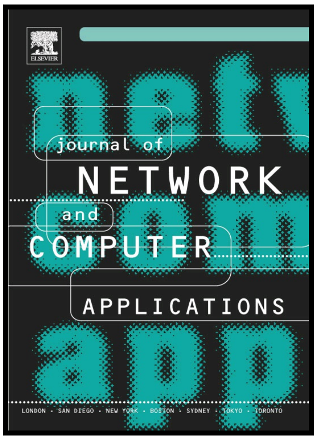

www.elsevier.com/locate/jnca

Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends

Alireza Sadeghi Milani, Nima Jafari Navimipour

PII: S1084-8045(16)30121-7
DOI: http://dx.doi.org/10.1016/j.jnca.2016.06.003
Reference: YJNCA1660

To appear in: *Journal of Network and Computer Applications*

Received date: 12 February 2016
Revised date: 2 May 2016
Accepted date: 4 June 2016

Cite this article as: Alireza Sadeghi Milani and Nima Jafari Navimipour, Loa balancing mechanisms and techniques in the cloud environments: Systemati literature review and future trends, *Journal of Network and Compute Applications,* http://dx.doi.org/10.1016/j.jnca.2016.06.003

Load balancing mechanisms and techniques in the cloud environments: systematic literature review and future trends

Alireza Sadeghi Milani, Nima Jafari Navimipour[1][2]*

Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

*Corresponding author. Tel.: +989144021694; fax: +984134203292. jafari@iaut.ac.ir

Abstract

With cloud computing, new services in information technology (IT) emerge from the convergence of business and technology perspectives which furnish users access to IT resources anytime and anywhere using pay-per-use fashion. Therefore, it should supply eminent functioning gain to the user and simultaneously ought to be advantageous for the cloud service provider. To accomplish this goal, many challenges have to be faced, where load balancing is one of them. The optimal selection of a resource for a particular job does not mean that the selected resource persists optimized for the whole execution of the job. The resource overloading/under-loading must be avoided which could be gained by appropriate load balancing mechanisms. However, to the best of our knowledge, despite the importance of load balancing techniques and mechanisms, there is not any comprehensive and systematic review about studying and analyzing its important techniques. Hence, this paper presents a systematic literature review of the existing load balancing techniques proposed so far. Detailed classifications have also been included based on different parameters which are relying upon the analysis of the existing techniques. Also, the advantages and disadvantages associated with several load balancing algorithms have been discussed and the important challenges of these algorithms are addressed so that more efficient load balancing techniques can be developed in future.

Keywords: Cloud computing, Load balancing, Task scheduling, Virtual machine

1. Introduction

Nowadays, with the rapid extension of the IT-based systems (Navimipour and Soltani 2016, Zareie and Jafari Navimipour 2016), many distribution systems such as the social networks (Sharif, Mahmazi et al. 2013, Mohammad Aghdam and Jafari Navimipour 2016), grid computing (Khanli and Razavi 2008, Navimipour and Khanli 2008, Navimipour, Rahmani et al. 2014, Souri and Navimipour 2014), cloud computing (Navimipour 2015, Navimipour and Milani 2015, Asghari and Navimipour 2016, Chiregi and Jafari Navimipour 2016, Milani and Navimipour 2016), Peer-to-Peer computing (Navimipour and Milani 2014), wireless networks (Navimipour and Rahmani 2009, Jafari Navimipour 2011, Jafari Navimipour and Es-Hagi 2011, Navimipour 2011, Navimipour, Shabestari et al. 2012), Expert Cloud (Jafari Navimipour, Masoud Rahmani et al. 2014, Navimipour 2015) and MapReduce (Navimipour and Khezr 2015) facilitate the data transfer and resource sharing (Navimipour and Milani 2015, Navimipour and Zareie 2015). Among them, cloud computing as a new concept to represent the cooperation among multiple computers and services via a network provides many powerful on-demand services to the users. Many new research and studies have been proposed in order to realize the concept of cloud computing, however, the fundamental idea of it is derived primarily from distributed computing and grid computing

[1] Neither the entire paper nor any part of its content has been published or has been accepted for publication elsewhere.

[2] It has not been submitted to any other journal.

(Cho, Tsai et al. 2014). The ordinary applications have been delivered online by cloud computing providers, which could access by a web browser as the software and data are stored on servers. Users no longer need the knowledge of the technology of the cloud that supports their computing requirements. Cloud computing provides a new supplement, delivery, and consumption model for IT services based on the Internet. It generally involves the provision of dynamically scalable and frequently virtualized resources as a service over the Internet (Chiregi and Navimipour 2016). It is a consequence of the ease-of-access to remote computing sites provided by the Internet (Li 2012). There are several types of cloud services available such as Software as a Service (SaaS) (Buyya, Broberg et al. 2011, F. Liu 2011, Abrishami and Naghibzadeh 2012, Celesti, Fazio et al. 2012, Jose Moura 2015), Platform as a Service (PaaS) (Marston, Li et al. 2011, Voorsluys, Broberg et al. 2011, Beloglazov, Abawajy et al. 2012, Jose Moura 2015), Infrastructure as a Service (IaaS) (Chang , Buyya, Broberg et al. 2011, Chou 2015, Jose Moura 2015) and Expert as a Service (EaaS) (Navin, Navimipour et al. 2014, Ashouraie, Jafari Navimipour et al. 2015, Jafari Navimipour, Rahmani et al. 2015, Navimipour, Navin et al. 2015, Navimipour, Rahmani et al. 2015). Furthermore, to the best suit of requirements of an application, the elasticity, flexibility and scalability have been offered to acquire or release resources varying configuration (Banerjee, Adhikari et al. 2015).

A workload or task abundantly overloads a resource in a computing environment. Therefore, the workload ought to be migrated to another resource. Consequently, three common operations need to be performed including load balancing, which checks the load on resources, resource discovery, which discovers another suitable resource and workload migration, which transfers the workload to the selected resource. These operations are taken over by three separate units, commonly called load balancing, resource discovery and process migration units, respectively. Evidently, a better performance could be obtained in the case of reduction of these operations (Arab and Sharifi 2014). Load balancing algorithms offer possibilities for increasing the performance of large-scale computing systems and applications since they are designed to redistribute the workloads over the components of the computing system in a way that guarantees to minimize response time, maximizing resource utilization and throughput, and avoiding the overload possibility (Daraghmi and Yuan 2015). To make better utilize of resources, an efficient load balancing solution needs to potentially reduce the resource over-provisioning (Nakai, Madeira et al. 2014). There are several models and techniques that offer efficient scheduling and load balancing such as statics and dynamics. Static mechanisms require prior knowledge of the environment and applications requirements. However, since the applications start executing, these models have no way to adapt to changes in the environment or requirements. In contrast, in dynamic mechanisms the load balancer monitors the environment and application requirements during run-time and attempts to make adjustments to redistribute the tasks and adjust the load as necessary (Mohamed, Al-Jaroodi et al. 2013).

Nevertheless, to the best of our knowledge, despite the importance of load balancing mechanisms in cloud environments, there is not any detailed and comprehensive systematic review of these mechanisms. Therefore, the purpose of this paper is to survey existing techniques, compares the differences between mentioned mechanisms, describes several popular load balancing mechanisms and outlines the types of challenges that could be addressed. We divided most of the introduced load balancing algorithms into two main categories, static and dynamic. To the best of our knowledge, this survey represents the first attempt to systematically examine load balancing with a specific focus on cloud computing. Briefly, the contributions of this paper are as follows:

- providing an overview of existing challenges in a range of problem domains associated with cloud computing that can be addressed using load balancing

- providing a systematic overview of the existing techniques for load balancing, and the manner in which these have been applied to cloud computing
- exploring the future challenges for cloud computing and the role that load balancing can play
- outlining the key areas where future research can improve the use of load balancing techniques in cloud computing

The rest of this paper is structured as follows. Section 2 discusses some related work. The research methodology is provided in Section 3. Section 4 discusses load balancing mechanisms in cloud computing and categorizes them, also presents the taxonomy and comparison of selected mechanisms. Section 5 maps out some validity threats. Section 6 discusses open issues. Finally, Section 7 concludes this paper.

2. Related work

Many types of research have been done in the field of cloud computing and general challenges including scheduling, resource provisioning and load balancing and etc. However, there is a little comprehensive research about cloud load balancing has been done yet. In this section, we refer to some papers that there are in the field of load balancing in cloud computing.

One of the significant surveys of the load balancing and job migration techniques in grid have presented by Rathore and Chana (2014). Also, it included a detailed classification based on different parameters which are depending on the analysis of the existing techniques. In their survey different parameters including research focus, contribution, compared model, strength, gap, future work and their suitability for usage in a dynamic grid environment has been analyzed. Furthermore, they proposed a new Load balancing technique, along with job migration and discussed to fulfill the existing research gaps. However, their load balancing survey was in the field of grid.

Kalra and Singh (2015) have presented a comparative analysis of various scheduling algorithms for cloud and grid environments based on five primary meta-heuristic techniques: Ant Colony Optimization (ACO), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), League Championship Algorithm (LCA) and BAT algorithm (Kalra and Singh 2015). They also presented a detailed comparison of various algorithms for each of these primary techniques. However, their survey was only limited to meta-heuristic techniques.

As another meta-heuristic survey to investigate existing resource allocation techniques for maximizing financial gains and minimizing the financial expenses of cloud users for IaaS in cloud computing environment, Madni, Latiff et al. (2016) have selected twenty-three studies that focused on the meta-heuristic algorithms for their research from 1954 to 2015. They compared these algorithms with traditional algorithms and concluded that there are numerous techniques to enhance the performance of meta-heuristics have been considered in these algorithms. Besides, to resolving resource allocation problems in cloud computing, none of the reviewed meta-heuristic algorithms is adequate to achieve distinctly superior performance than other algorithms. However, this survey similar to the previous one is limited to meta-heuristic techniques. As another review to study the various concepts and scheduling algorithms used for the on-demand Grid as a service Cloud (GaaS).

Abdulhamid, Abd Latiff et al. (2014) have evaluated selected scheduling algorithms based on the parameters such as load balancing, energy consumption, makespan and etc. They concluded that none of the reviewed scheduling algorithms fulfill the entire scheduling parameter requirements. However, their review evaluated task scheduling algorithms and also there is a gap for discussing the open issue and challenges in their review.

Based on requirements specified in service level agreement, Katyal and Mishra (2013) have reviewed various load balancing schemes in a different cloud environment. They categorized the algorithms based on a cloud environment, the spatial distribution of nodes and Task dependencies and discussed advantages and disadvantages of each category. However, they evaluation only included the main categorize and did not use load balancing parameters in order to evaluate the application of each selected algorithm.

It is important to point out that none of these surveys present a pure systematic literature-based review of the existing load balancing techniques with a discussion on their categorization, future challenges and the crucial role that load balancing could have in a cloud environment. In this paper, we formalize three question in the next section in order to select most significant studies for review (both dynamic and hybrid approaches) and then underline the importance of load balancing mechanisms, current challenges, and future trends in cloud environments by answering to each of these questions.

3. Research methodology

To advance our understanding of load balancing techniques, this section has been carried out according to the guideline for systematic literature review (SLR) proposed by (Kitchenham 2004) with a specific focus on research related to load balancing mechanisms in the cloud. An SLR is a research method originating from the field of medicine (Kitchenham 2004) which provides a repeatable research method and should supply sufficient detail to be replicated by other researchers (Kupiainen, Mäntylä et al. 2015, Charband and Navimipour 2016, Jafari Navimipour and Charband 2016). In terms of lead to detailed answer within necessity of load balancing in computing environments, we developed three research questions to address the key concerns of load balancing in Cloud Computing. In the following section, we formalize these questions.

3.1. Question formalization

The goal of this section was to name the most relevant issues and challenges in a cloud-based load balancing, including response time, vulnerabilities, underloading, overloading, costs, and possible balancing solutions. This research effort will thus aim to address the following research questions:

RQ1: what is the importance of load balancing with usage growth of cloud systems?
This question aims at the number of cloud load balancing studies have been published over time, to underline the importance of load balancing along with increasing cloud usage.

RQ2: How much are existing load balancing approaches meet the main load balancing metrics?
The main purpose of this question is to evaluate existing load balancing approaches based on primary load balancing metrics in a cloud environment.

RQ3: Which problems and solutions were identified with regard to load balancing for future trends?
The objective of this question is to understand the role of load balancing in Cloud Computing, identifying its challenges and techniques used to ensure Quality-of-Service (QoS) in the environment.

A process as such can lead to detailed answers within the scope of this paper. After identifying the need for research, research questions were formulated a review protocol for our study. This protocol development has different stages, such as search query, selection of source and criteria, quality assessment criteria, data extraction and data synthesis strategy.

3.2. Search query

Search strings were developed for academic databases and inclusion and exclusion criteria, by defining keywords. Search strings are defined by identifying synonyms and alternative spellings for each of the question components and associate them by utilizing the Boolean OR and Boolean AND (Soltani and Navimipour 2016). A search string has been defined by the selection of the most befitting keywords in terms of providing our subject. Hence, four keywords have been selected: “load balancing”, “cloud”, “distributed”, “task scheduling”. After different steps and utilizing the outcomes of our initial analysis as a pilot to examine the coverage of the outcomes, the query was defined. Namely, we refined the query string to add more keywords, in the case of studies in our pilot was not retrieved by the primary query for example “load balancing” OR “workload balancing” AND “cloud”, “task migration” OR “virtual machine migration”. In order to expand the scope as far as possible, the search string was applied to titles, abstract and body of the studies. The search was conducted in December 2015, with a specified time range from 2010 to 2015.

3.3. Selection of sources

Primary journal articles and conference papers have been selected for our search query. We subsequently classified and analyzed these publishers in order to extract relevant results. Google scholar, Scopus, and Web of Science were adapted as our data source. Consequently, the search process covered articles available in five of the most reliable databases that are technically and scientifically peer-reviewed: ACM Digital Library, IEEE Xplore, Springer Link, Science Direct, and Elsevier among others.

3.4. Selection criteria

To be qualified for inclusion in this review a quality assessment checklist (QAC) based on Kitchenham, Pearl Brereton et al. (2009) is developed to assess only individual papers from peer-reviewed journals published from 2010 to 2015. The checklist includes the following questions (Kitchenham, Pearl Brereton et al. 2009): (a) Does the research paper clearly specify the research methodology? (b) Is the research methodology appropriate for the problem under consideration? (c) Is the analysis of study properly done? If the study fulfills assessment criteria then it is filled with 'yes'. Table 1 summarizes the inclusion–exclusion criteria for our review protocol. Furthermore, in this paper due to the necessity of cloud environment for optimized load balancing, we excluded static based mechanisms. In section 4, we present an explanation for this.

Table 1. Summary of the inclusion-exclusion criteria for review protocol

Criterion	Rational
Inclusion1 A study that clearly described how the mentioned load balancing technique(s) could be applied and assisted in cloud computing environment were selected.	We want to identify how load balancing affect computing in the cloud, thus, we need articles that directly proposed load balancing in the cloud or indirectly proposed them from a service provisioning perspective.
Inclusion2 A study that is developed by either of academics or practitioners.	Both academic and industrial solutions are relevant to this study.
Inclusion3 A study that is published in cloud computing field.	Cloud computing is our reference field.
Inclusion4 A study that is peer-reviewed.	A peer-reviewed paper guarantees a certain level of quality and contains a reasonable amount of content.
Inclusion5 A study that is written in English.	For feasibility reasons papers written in other languages than English are excluded.
Exclusion1 A study that includes journal papers only.	Conference papers, masters and doctoral dissertations, textbooks, editorial notes, and unpublished working papers were excluded, as academics and practitioners alike most often use journals to acquire information and disseminate new findings.
Exclusion2 A study that does not focus on load	The focus of this paper is only on studies that present load balancing

balancing techniques in the field of cloud computing environments.	techniques specifically in the cloud.

3.5. Quality assessment and data extraction

The phase of data extraction summarize the data from the selected studies for further analysis. We identified a total of 726 studies. Primarily, we read abstracts and search keywords plus concepts that reflect the contribution of the paper. Therefore, insufficient abstracts were eliminated. Then, for the remains, the full body of each paper was reviewed, those that were not related to the application of our specific topic were also eliminated. After filtering these studies according to inclusion/ exclusion criteria and QAC, 18 publications were identified as a primary study for review. An overview of the used process to identify the articles in this study is illustrated in Fig. 1.

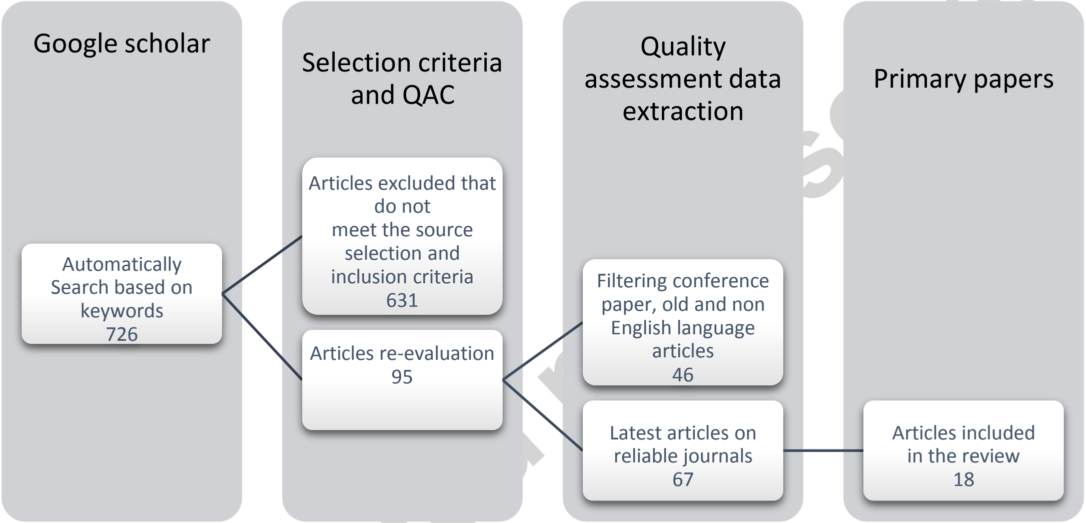
Google scholar
Selection criteria and QAC
Quality assessment data extraction
Primary papers
Automatically Search based on keywords 726
Articles excluded that do not meet the source selection and inclusion criteria 631
Articles re-evaluation 95
Filtering conference paper, old and non English language articles 46
Latest articles on reliable journals 67
Articles included in the review 18

Figure 1. An overview of the utilized article identification process

Fig. 2 demonstrates the distribution of the selected primary articles over time. As can be seen, there is a significant rise in the number of papers on the scope of cloud load balancing from 2010 to 2015; also, most of the selected papers were published in 2015. Due to our first formalization question (RQ1), it distinctly outlines the importance of load balancing and necessity of new and improved load balancing mechanisms along with the rise in the utilization of cloud systems. As illustrated in Fig. 2(a), for the sake of the growing interest of load balancing among researchers we included conference papers, however, we excluded them as mentioned in Table 1, as academics and practitioners alike most often use journals to acquire information and disseminate new findings.

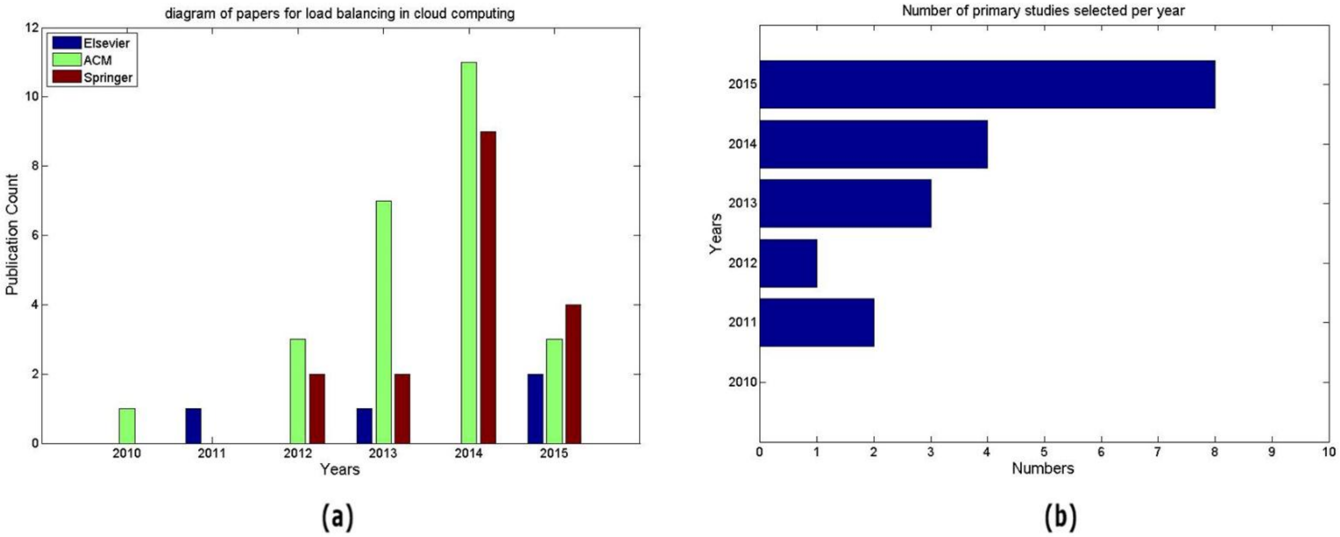

Figure 2. (a) Distribution of the articles over time including Elsevier, Springer, and ACM. (b) Number of primarily selected studies per year

4. Load balancing strategies

In order to answer RQ2 and RQ3, we thoroughly reviewed the selected studies and identified the most frequently addressed load balancing in cloud computing. Load balancing algorithms based on the system topology and availability of information about resources categorized as static and dynamic. Static algorithms do not depend on the current state of the system, prior knowledge of the system is required, such as job resource requirements, communication time, the processing power of system nodes, memory and storage devices capacity and so on. Static load balancing algorithms use the polling approach, also called the Round Robin approach. This method is simple and uses fewer resources, but is generally unable to detect the attached server, resulting in annexation or uneven distribution (Chen, Chen et al. 2016). Therefore, the main drawback of static load balancing algorithms is that the current state of the system is not considered when making the decisions and therefore it is not a suitable approach in systems such as distributed systems which most states of the system changes dynamically. Whereas dynamic algorithms are based on the current state of the system and used to deal with unpredictable processing loads. Based on network storage virtualization, the host and storage devices use fiber channel switches that link together, and all virtualization requests return to the network storage device (Chen, Chen et al. 2016). In dynamic approaches, tasks can move dynamically from an overloaded node to an underloaded one and this is the main advantage of dynamic load balancing algorithms which can change continuously according to the current state of the system. However designing and implementing a dynamic load balancing algorithm is much more complicated and harder than finding a static solution, but through dynamic mechanisms we can gain a higher performance and have more accurate and efficient solutions (Ali M. Alakeel 2010, Kanakala, Reddy et al. 2015). Static algorithms work properly if only nodes have a low variation in the load. Hence, these algorithms are not desirable for cloud environments because of varying load at different times (L.D and Venkata Krishna 2013). As stated in section 3.4 in this paper due to the necessity of cloud environment for optimized load balancing, we excluded static based mechanisms. The dynamic nature of cloud computing environment, the processing time for tasks will lead to an uneven distribution of the server, needs dynamic algorithms for efficient scheduling and load balancing among nodes (Chen, Chen et al. 2016). Additionally, according to our search query also considering limitation to published years there were not any significant research with the subject of static mechanism in large computing environments, instead, we describe most popular hybrid mechanism proposed in the cloud in section 4.2.

4.1. Load balancing metrics

Load balancing is applied to distribute the dynamic local workload equally across all the resources or virtual machines. It supports to attain resource utilization and high user satisfaction ratio together with assuring a competent and fair allocation of every resource. An appropriate load balancing aids in minimizing resource consumption, maximizing scalability, avoiding bottlenecks and over-provisioning etc. The important qualitative metrics for load balancing in cloud computing are discussed as follows:

Response time: It measures the total time that the system takes to serve a submitted request (L.D and Venkata Krishna 2013, Xu, Pang et al. 2013, Daraghmi and Yuan 2015).

Scalability: It determines how the system is capable of accomplishing load balancing algorithm with a restricted number of machines or hosts (Lu, Xie et al. 2011, Nakai, Madeira et al. 2014).

Resource Utilization: It is the degree to which the resources of the system are utilized. A desirable load balancing algorithm provides maximum resource utilization (Fang, Wang et al. 2010, Banerjee, Adhikari et al. 2015, Wang, Chen et al. 2015).

Throughput: The rate at which a node in the system sends or receives data. Simply put, it is defined as the number of nodes that change its status to "complete" in a time unit. For better performance of the cloud, a high throughput is required (Maguluri, Srikant et al. 2012, L.D and Venkata Krishna 2013).

Migration time: The time taken in the migration of a task from one machine to any other machine in the cloud system. It should be minimized for improving the performance of the cloud system (Fang, Wang et al. 2010, Ramezani, Lu et al. 2013).

Makespan: Makespan specifies the maximum completion time or time when the resources are allocated to the users (Abdulhamid and Latiff 2014).

Fault tolerant: It is the capability of the algorithm to perform uniformly and properly even in conditions of failure at any arbitrary node in the system (Voorsluys, Broberg et al. 2009).

The degree of imbalance: It measures the imbalance among virtual machines (Abdullahi, Ngadi et al. 2016).

Performance: It represents the effectiveness of the system after performing load balancing. If all the above parameters are satisfied optimally then it will highly improve the performance of the system (Zhang and Zhang 2010, Cho, Tsai et al. 2014, Wang, Chen et al. 2015).

4.2. Review of the dynamic load balancing technique

In this section, we first describe the dynamic load balancing mechanisms and their basic properties. Second, we discuss most popular dynamic load balancing mechanisms. Finally, the discussed dynamic mechanisms are compared and summarized in Section 4.2.2.

Dynamic load balancing algorithms have more practical value than static ones. Most of the Dynamic load balancing algorithms rely on a combination of knowledge and run-time properties. Knowledge is established by prior collected information about the nodes in the cloud and run-time properties gathered as processing the task by nodes. These algorithms assign the tasks to a node and dynamically reassign them to another node, to improve the overall performance, based on the attributes gathered and calculated. The next section provides a review and survey on several important dynamic load balancing mechanisms.

4.2.1. Dynamic load balancing techniques

One of the techniques to accomplish system load balancing in the cloud environment is a live virtual machine (VM). It achieves by transferring an active VM from one physical host to another (Ramezani, Lu et al. 2013). A task-based Load Balancing method using particle swarm optimization have been proposed by Ramezani, Lu et al. (2013) to achieve load balancing by only migrating extra tasks from an overloaded VM rather than migrating the entirely overloaded VM. The authors proposed a model to migrate these extra tasks to the new host VMs by utilizing particle swarm optimization. To evaluate, they extended the cloud simulator (CloudSim) package and used particle swarm optimization as its task scheduling model. The simulation results show that proposed method reduced the time consumption of the load balancing process and energy consumption compared to other traditional methods. Furthermore, the method reduced downtime memory, memory usage and cost consumption because there is no need to pause VM during migration time, and much less idle capacity in the host physical machine (PM) is required. However, it only migrated arrival independent tasks from an overloaded VM to another homogenous VM.

Another solution for load balancing is proposed by L.D and Venkata Krishna (2013), which is an algorithm named honey bee behavior inspired load balancing to achieve well-balanced load across virtual machines. The tasks that require being balanced are proposed as bees and the VMs are the food sources while low load machines are represented as bees' destinations. Loading of a task to a VM is represented as a bee foraging a food source. Once a VM is overloaded the task will be scheduled to an underloaded VM. Removed task updates the remaining tasks about the VM status. Hence, based on the load and availability of the VMs, this updating give a clear idea in deciding which task should be assigned to which VM. Three possible priorities were defined for tasks which affect the allocation decisions. The method improved the overall throughput and reducing the amount of time a task has to wait in a queue of the VM. Thus, it reduced the response time and makespan. However, it considered tasks as independent and may not be classified as a highly scalable mechanism.

As another solution by utilizing honey bee inspired load balancing Remesh Babu and Samuel (2016) have proposed the honey bees foraging behavior for load balancing across VMs and reschedules the cloudlets into underloaded VMs. In their method, the tasks removed from overloaded VMs have treated as honeybees and under loaded VMs are the food sources. The proposed method works in four different steps. VM current load calculation, load balancing and scheduling decision, VM grouping and task scheduling. The experimental result showed improvement in the QoS delivered to the customers, minimize makespan and VM migrations. However, it still suffers from low scalability.

In 2015, another VM live migration technique for load balancing was proposed by Gutierrez-Garcia and Ramirez-Nafarrate (2015). A collaborative agent-based problem solving technique enhanced with VM migration policies to enable agents to establish preferences over in case of VMs migrating, VM acceptance policies to determine whether to participate in the bid for hosting a given VM from a host, VM migration heuristics to determine which VM should be migrated and its destination host, a set of load balancing heuristics of the front-end agent to select the initial hosts of VMs and a distributed problem solving technique to evenly distribute loads across hosts (Gutierrez-Garcia and Ramirez-Nafarrate 2015). The results showed that agents collaboration could efficiently balance the loads in a distributed manner with a high performance comparable to commercial solutions, namely Red Hat (Gutierrez-Garcia and Ramirez-Nafarrate 2015) while fewer VMs migrating rate. However, it is a centralized approach which is not fully scalable and still suffers from high VM migration overhead.

Also, the Quality of Service (QoS) of a cloud service providers is an important research field which covers several critical issues such as efficient load balancing. Banerjee, Adhikari et al. (2015) presented a cloudlet allocation strategy with suitable load balancing technique in which allocate a cloudlet from the sorted cloudlet list to a VM whose load capacity is maximum among all the VMs present in the VM list. Therefore, that assists in distributing the cloudlets to the virtual machines evenly which makes the system more active and balanced. Hence, the QoS, resource utilization, completion time and makespan of the overall system have been improved in comparison with the other existing cloudlet allocation policies. However, it suffers from high makespan and low scalability.

Wang, Lin et al. (2012) have proposed a market-based control method, called MBA, to address load balancing in cloud database via data allocation and dynamic migration. MBA was a market-based control method in which database nodes are considered as traders in a market and certain market rules (Niu, Cai et al. 2008) are utilized to decide data allocation and migration, intelligently. This method contains two types of agents, i.e., data trading agent, responsible for the real time tracking of query load on a cloud database node and determining the bid or ask price for the cloud database node and data auctioning agent which is responsible for which traders can buy and which traders can sell. Therefore, the system can achieve load balance efficiently and adapt to the changes over time effectively. Experimental results showed that the MBA improved system performance in terms of average query response time and fairness. However, it may take several rounds to satisfy the buyer’s need.

Mohamed, Al-Jaroodi et al. (2013) have introduced a fast and efficient dual direction file transform protocol (FTP) technique for downloading large files from replicated cloud data servers. The proposed technique offers efficient load balancing within heterogeneous servers with minimal overhead. It uses the concept of processing the files in two different directions. Thus, if two replicas of a file exist on two servers, one server sends starting blocks at the beginning of the file, while the second one starts it from the end. They will proceed till meeting each others. Then, the client will ask them both to stop. This provides automatic load balancing granting each server to work rely on its conditions yet both finishing at the same time (Mohamed, Al-Jaroodi et al. 2013). As a result, network utilization could be maximized whereas resources and current loads vary dynamically, load balancing remains efficient. However, download time among different partitions and servers is still high. Also, the existence of full replicas on each of the cloud server is required.

Nakai, Madeira et al. (2014) have proposed a load balancing mechanism based on the reservation policy to distribute requests among the replicated servers. The method allows overloaded servers to reserve the capacity of remote servers before transferring requests to them. The reserved capacity for a server could not be shared with any other servers. By limiting the amount of transferring load from each server to the others, the overload of remote servers is prevented. Hence, service response time is reduced. Also, they described a middleware to support this mechanism by administering the trade of resources among a set of servers, to share their spare capacity, transparently. The simulation results demonstrated that the technique presented better response times, low number of discarded requests, better resilience to abrupt load changes, better balances, and lower messaging overhead. However, although the number of discarded requests is low but still occurs.

Furthermore, De Falco, Laskowski et al. (2015) have proposed a load balancing algorithm for clusters of multicore processors. The nature-inspired optimization approach is utilized to discover the best tasks as candidates for migration, periodically. Also, for the best selection of computing nodes to receive the migrating tasks. Their algorithm is an improved version of the Extremal optimization (EO) based load balancing algorithm in distributed applications (Boettcher and Percus 1999, De Falco, Laskowski et al. 2015), which is modified by replacing the fully random computing node selection with the stochastic selection in which the probability is guided by some knowledge of the problem (De Falco, Laskowski et al. 2015). The algorithm was compared against a greedy fully deterministic approach, a genetic algorithm, and an EO-based algorithm. The comparison showed the improvement in the quality of load balancing with extreme optimization. However, the method suffers from low scalability and high response time for cloud environments.

To eliminate the system bottlenecks and resource utilization optimization, Wang, Chen et al. (2015) have proposed a framework of workload balancing and resource management for Swift, a widely utilized and distributed storage system on the cloud. In this framework, they designed workload monitoring and analysis algorithms for discovering overloaded and underloaded nodes in the cluster. To balance the workload among those nodes, split, merge and pair algorithms are implemented to regulate physical machines while resource reallocates algorithm is designed to regulate virtual machines on the cloud. Experimental results demonstrated that the proposed framework is lightweight and required no source code modification of the guest operating system and storage system. In addition, the proposed framework tunes the system performance improves computation of resource utilization and also ensures the reliability of the storage system through adopting live virtual machine migration method as its migration method. However, low resource utilization might cause high costs of hardware resources and it cannot always improve the system performance.

As we know, the performance of rebalancing is significantly affected by the load balancing technical factors. Therefore, Daraghmi and Yuan (2015) have proposed an approach to improve the performance of load balancing algorithms by taking into account both the load balancing technical factors and the structure of the network to execute the algorithm. They have designed an improved load balancing algorithm that effectively executed among the constructed overlay network, namely, the functional small world (FSW) which facilitates efficient load-balancing in heterogeneous systems, where nodes consider the capacity and calculate the average effective load. Proposed approach strove to balance loads of nodes, increased the system throughput, decreased both the response time and communication overhead, and deteriorated the demanded movements cost as much as possible. However, the method resource utilization is low. Also, scalability has not matched the requirement of cloud environments.

4.2.2. Summary of the dynamic load balancing mechanisms

In prior section, we described most popular dynamical mechanisms for load balancing in the cloud. The discussions provide important features about dynamical load balancing mechanisms. With these mechanisms system performance in terms of response time significantly improved. Also, the efficient utilization of resources has been maximized. Furthermore, flexibility and scalability of cloud would be guaranteed and overall throughput is increased. In addition, costs for both customer and provider are reduced. However, since the decision-making for the selection of resources due to transfer load information among resources is essential, the run-time overhead could be a major issue, cost and time consuming are crucial and memory utilization is high. Also, the complexity of dynamic load balancing mechanisms is another issue. Table 2 provides a brief summary of popular dynamic load balancing mechanism and their main properties.

Table 2. Popular dynamic load balancing mechanisms and their properties

Reference	Technique	Main idea	Advantages	Disadvantages
Ramezani, Lu et al. (2013)	Task-based system load balancing method using particle swarm optimization (TBSLB-PSO)	Using live VM migration techniques.	• Low task execution time. • Low task transferring time. • Low task execution cost. • Low memory usage.	• Homogeneous virtual machines. • Independent tasks.
L.D and Venkata Krishna (2013)	Honey bee behavior inspired load balancing (HBB-LB)	Using the foraging behavior of honey bees.	• Low response time. • Low makespan.	• Independent tasks. • Low scalability.
Remesh Babu and Samuel (2016)	Enhanced bee colony algorithm	Using the foraging behavior of honey bees.	• Low makespan • Low VM migration	• Low scalability
Gutierrez-Garcia and Ramirez-Nafarrate (2015)	Agent-based load balancing	Using the agent adopting technique based on live VM migration enhanced with VM migration policies, VM acceptance policies, VM migration heuristics, and a set of load balancing heuristics.	• No resource monitoring. • Heterogeneous server and virtual machines.	• High migration overhead. • Low scalability.
Banerjee, Adhikari et al. (2015)	New cloudlet allocation policy	Using the cloudlet allocation policy with load balancing.	• Improved makespan. • Better Quality of service. • High resource utilization. • Equal distribution of load to each VM.	• High makespan. • Low scalability.
Wang, Lin et al. (2012)	Market-based approach (MBA)	Using The MBA (Market, Bid and Ask) method based on double auction mechanism.	• High flexible and scalable database. • Low response time.	• High rounds to satisfy the buyer's need.
Mohamed, Al-Jaroodi et al. (2013)	Dual-direction file transform protocol (DDFTP)	Using the concept of processing the files in two different directions.	• Low monitoring. • Low network overhead. • Automatic ability to adjust to the changing conditions. • Minimum need for server's information.	• High Download time among different partitions and servers. • Requires the existence of full replicas of the data set on each of the cloud servers.
Nakai, Madeira et al. (2014)	Server-based load balancing policy	Using the Protocol for reservation policy of remote resources.	• Low response time. • Low network overhead. • High scalability.	• Discarded requests.
De Falco, Laskowski et al. (2015)	Extremal optimization (EO) processor load balancing	Using the nature-inspired optimization technique to the moderate computational complexity and small memory requirements for clusters of multicore processors.	• Parallel speedup. • High load balancing. • Low application execution time. • A low number of migrations.	• Low scalability. • High response time.

Wang, Chen et al. (2015)	Dynamic workload-aware balancing framework	Using the protocol based on virtualization technology for workload balancing with a split, merge and pair algorithms in cloud swift.	• No system bottlenecks. • High resource utilization. • No source code change. • Improved workload balancing in physical machines. • Tuning system performance.	• Low resource utilization. • High costs of the hardware resource.
Daraghmi and Yuan (2015)	Functional small world (FSW)	Using the concept of small world network and the node services scope, then.	• Structural and the technical load-balancing factors. • Low response time. • High-throughput. • Low task transferring time.	• Low resource utilization. • Low scalability for cloud environments.

4.3. *Hybrid load balancing techniques*

In this section, first, we present a basic description and overview of hybrid mechanism, and then some of the popular hybrid techniques of load balancing are provided. Finally, the provided techniques are compared and summarized.

Hybrid mechanisms have been proposed to surmount the presented drawbacks of dynamic and static load balancing techniques by compounding them while retaining each mechanism's benefits and advantages. These mechanisms gain by the efficiency of each compounding mechanism while overcoming their inherited drawbacks. In the next section, several hybrid mechanisms and their main features are discussed. These studies have been selected by the methodology that we mentioned in section 3.

4.3.1. *Review of the hybrid load balancing techniques*

Lai and Yu (2011) have proposed a scalable multi-attribute hybrid overlay for information sharing and resource discovery in cloud environments. Additionally, the proposed hybrid overlay integrates a structured peer-to-peer system with an unstructured one to support complex queries. Each node is regarded as the neighbor of other nodes in the same resource group. Besides, each node has a threshold of accepted queries, and the threshold is defined based on the capability of the node. The method makes use of transferring the redundant connections and migrating queries to low-load neighboring nodes for load balancing (Lai and Yu 2011). Experimental results showed that except for a small number of nodes, most of the high-load nodes could transfer their queries to other low-load nodes to achieve load balance. However, this method suffered excessive delay when transferring redundant connections to other low loading neighbors. Also, there are small failures of transfer high load queries to other nodes.

To resolve the VM scheduling problem, Cho, Tsai et al. (2014) have proposed the combination of ant colony optimization (ACO) and particle swarm optimization. The method utilizes historical information to anticipate the workload of new requests to adapt to dynamic environments without additional task information. Furthermore, to minimize the computing time of the scheduling procedure, the pre-reject step rejects requests that could not be satisfied before scheduling (Cho, Tsai et al. 2014). Experimental results showed that it is faster than traditional ant colony optimization and, in most cases, the method kept loads more balanced than the other approaches, and shorter makespan in single scheduling. However, the method has proposed on Homogeneous servers and costs was not included for clients and provider.

To deliberate the behavior of resources capacities, running processes, and utilize patterns, resource discovery and load balancing decisions are made cooperatively (Arab and Sharifi 2014). Therefore, Arab and Sharifi (2014) have proposed a model for this communication among resource discovery and load balancing units in a computing system. According to their model, resource discovery and load balancing units are adapted into two separate units. Each node is equipped with a load balancing unit as well as a resource discovery unit. By exchanging messages, load balancing units have thorough knowledge about the status of resources. Therefore, more accurate decisions could be made by resource discovery units. As they have proposed, the load balancing mechanism is in charge for the extraction of processes' behavior and it can utilize several techniques proposed in (Dodonov and de Mello 2010). The model reduces coupling between units. Therefore, along with its scalability on size, the model provides a high level of scalability. However, response time is low and there is an increase in resource discovery time.

A hybrid scheme of task scheduling and load balancing has proposed by Liu, Zhang et al. (2015). The method comprises three algorithms, including on-Demand scheduling, Querying and Migrating Task (QMT), and Staged Task Migration (STM). QMT is designed to keep the workload balanced, when a slave has a heavy load, the master will apprise it and designate the last dispatched task to another slave with the low load (Liu, Zhang et al. 2015). Results demonstrated that QMT and STM are effective for the independent task and dependent-task scheduling, respectively. However, scheduling and transmission time are high.

To distribute load balancing in dynamic and scalable systems, Lu, Xie et al. (2011) have proposed the class of algorithms namely, Join-Idle-Queue (JIQ). The decoupling discovery of lightly loaded servers from job assignment is the fundamental idea. The method has been defined to solve two problems. To solve the primary problem of assigning jobs to processors with load balancing, they first solve the secondary problem of assigning idle processors to dispatchers, which takes place in the reverse direction. While the reduction of average queue length at each processor is the primary concentration problem, the secondary problem concentrates the availability of idle processors at each dispatcher. Unlike algorithms such as Power-of-Two (Lu, Xie et al. 2011), this method obtains no communication overhead. It outperforms the state-of-the-art SQ algorithm (Lu, Xie et al. 2011) in terms of response time at the servers. The overall complexity of the method is no greater than that of SQ. however, it still suffers from high complexity.

To overcome server respond failures while a large number of users attempt to access cloud services, Chen, Chen et al. (2016) have proposed a dynamic Cloud load balancing (CLB) architecture takes into consideration both server processing power and computer loading, thus making it less likely that a server will be unable to handle excessive computational requirements. They also designed a cloud load balancing algorithm, capable of applying to both virtual web servers and physical servers, for monitoring platforms in order to obtain each server loading, computing power, and the priority service value. The results showed that cloud server performance based on proposed architecture can balance the loading with highly scalable performance. However, the average response time increases with the number of connections for both physical and virtual web servers.

Finally, Naha and Othman (2014) have proposed the combination of throttled and round robin VM load balancing policies with a performance optimized and service proximity based routing service brokering algorithm based on Wickremasinghe, Calheiros et al. (2010). They considered existing algorithm in a different simulation environment. The closest data center brokering policy chooses the data center closest to the user's region. However, optimized response time-brokering policy selects the nearest available data center based on network latency. Round robin load-balancing policy balances the user request load in ordinary round robin fashion. Conversely, the throttled load-balancing policies maintain a table for all the available VMs (Naha and Othman 2014). Results demonstrated that the combination of the closest data center brokering policy and throttled load balancing policy algorithm required the minimum processing time and the grouping of optimize response time-brokering policy and throttled load-balancing policy led to the lowest response time. However, the execution time for algorithms is high and performance needs improvement.

4.3.2. *Summary of hybrid load balancing mechanisms*

Hybrid mechanisms further reduce the response time, offer some efficient usage of resources and are more scalable. Moreover, the combination with other techniques provides further reducing in computing time. But, a major drawback is the inability to provide non-complex techniques. Table 3 provides the summary of main properties of hybrid mechanisms.

Table 3. Popular hybrid load balancing mechanisms and their properties

Reference	Technique	Main idea	Advantages	Disadvantages
Lai and Yu (2011)	Multi-attribute hybrid overlay (MAHO)	Using the vector of different attributes as key parameters to reserve resources.	• High Scalability. • High performance. • Low redundant connections. • High fault tolerance	• Failure of transferring high load. • High delay for transferring redundant connections.
Cho, Tsai et al. (2014)	Ant colony optimization with particle swarm (ACOPS)	Using the ACO-based with particle swarm VM scheduling algorithm.	• High resource utilization • Low computing time under high load. • Low response time.	• Homogeneous servers. • High costs.
Arab and Sharifi (2014)	Communication between resource discovery and load balancing	Using the model for communication between resource discovery and load balancing mechanisms.	• High scalability. • No needs for information of another node.	• High resource discovery time. • Propagation of policies.
Liu, Zhang et al. (2015)	Hybrid scheme of task scheduling and load balancing (DeMS)	Using the hybrid scheme of task scheduling and load balancing based on master slaves.	• Low response time. • Independent and dependent tasks scheduling.	• High scheduling and transmission time.
Lu, Xie et al. (2011)	Join-idle-queue (JIQ)	Using the distributed algorithm adaptable to cloud computing by eliminating the inter-node communications.	• Low communication overhead. • High scalability. • Low response time.	• High complexity. • Homogeneous resource.
Chen, Chen et al. (2016)	Cloud load balancing (CLB)	Using five levels architecture along with a load balancing algorithm for monitoring platform to obtain server loading power consuming and priority.	• High Scalability.	• High response time.
Naha and Othman (2014)	Brokering and load balancing combination	Using a combination of two load balancing algorithms and two brokering algorithms.	• Low response time • Low processing time	• High execution time • Low performance

4.4. Discussion

This section provides a comparative analysis of various load balancing metrics in cloud computing as shown in Table 4 and Figure 3. After the comprehensive review of the selected techniques proposed for the load balancing in cloud computing. It can be seen that the prospects of the hybrid techniques perform better (see Table 4 and Figure 4) in this area. The hybrid algorithms work both in favor of cloud user and cloud developer in which proposed approaches paid more attention to response time and scalability. In the cloud user perspective, makespan, response time are very important, which are used as significant metrics for load balancing in cloud computing. In addition to improving response time, the proportion of scalability in proposed hybrid techniques has been heightened to 33% in comparison with dynamic techniques which is 9% (Figure 4). This is an important point for developing cloud based systems which are highly growing nowadays. However, in terms of meeting proper makespan as shown in Figure 4, it is observed that dynamic techniques have more tending to make improvement in this aspect as this technique with a high percentage of response time (39%) and makespan (16%) keep cloud user at highest priority for developing new techniques. The chief drawback of the selected hybrid techniques is migration time which its proportion in this techniques has been diminished, unlike the selected dynamic techniques with the proportion of 23%. There could be some reasons for this, more likely new studies have no desire to make improvement for this aspect and prefer to concentrate on other metrics which along with the growth of could have required new and more efficient approaches. Furthermore, efficient resource utilization could enhance the time, cost and energy consumption and under-utilization of resources. As shown in Figure 4 percentage of Resource utilization metric for both techniques persists at the fair range. Overall, as demonstrated in Figure 3, response time (39%), scalability (16%) and migration time (16%) are the primary metrics have ameliorated in recent years.

Table 4. Load balancing metrics in reviewed techniques

Reference	Response time	Makespan	Scalability	Resource utilization	Migration Time	Throughput
Ramezani, Lu et al. (2013)	●			●		
L.D and Venkata Krishna (2013)	●	●				
Remesh Babu and Samuel (2016)		●			●	
Gutierrez-Garcia and Ramirez-Nafarrate (2015)	●					
Banerjee, Adhikari et al. (2015)		●		●		
Wang, Lin et al. (2012)	●		●			
Mohamed, Al-Jaroodi et al. (2013)					●	
Nakai, Madeira et al. (2014)	●		●		●	
De Falco, Laskowski et al. (2015)	●				●	
Wang, Chen et al. (2015)	●			●		
Daraghmi and Yuan (2015)	●				●	●
Lai and Yu (2011)			●			●
Cho, Tsai et al. (2014)	●			●		
Arab and Sharifi (2014)			●			
Liu, Zhang et al. (2015)	●					
Lu, Xie et al. (2011)	●		●			
Chen, Chen et al. (2016)						
Naha and Othman (2014)	●					

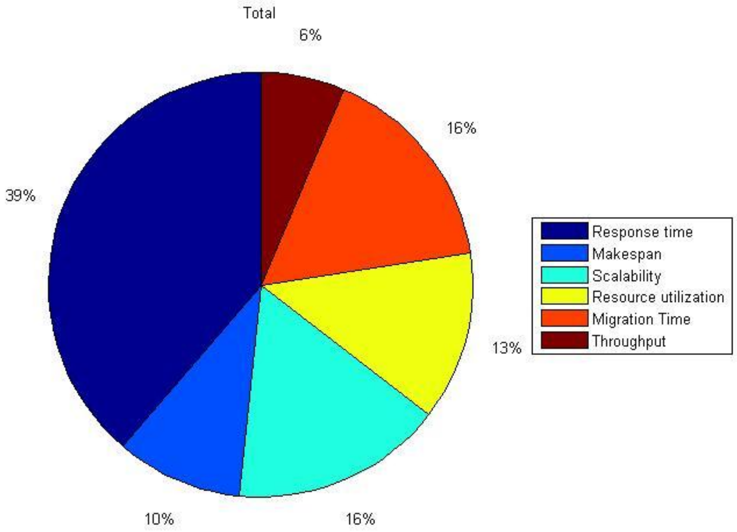

Figure 3. Percentage of load balancing metrics in reviewed techniques

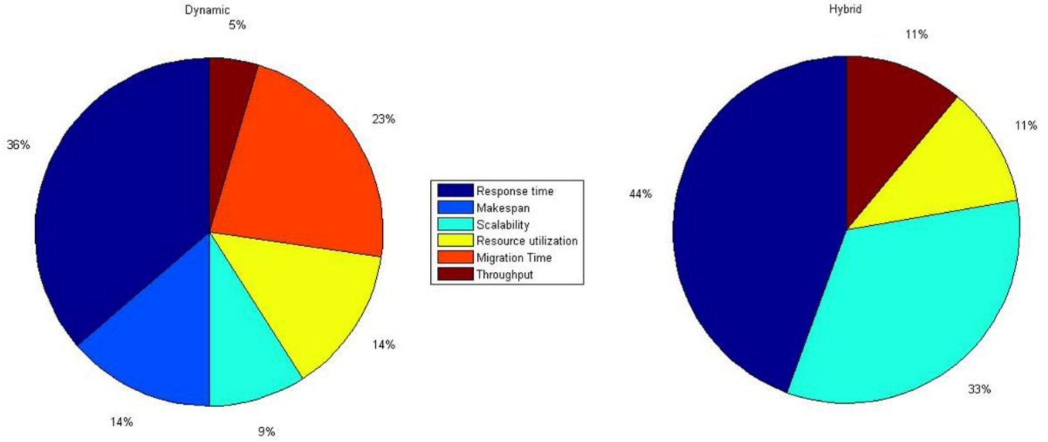

Figure 4. Percentage of load balancing metrics for dynamic and hybrid techniques

5. Threats of validity

We attempted to conduct this systematic review as rigorously as possible. However, it might have still endured from several validity threats. Hence, future efforts, to interpret or directly utilizing the reviewed or conclusions in this systematic review should bear in mind these limitations:

- *Research scope:* The application of the load balancing in cloud environments have been covered in various sources such as academic publications, editorial notes, technical reports and web pages, etc. Specifically, we have omitted articles published in national journals and conferences. Also, articles that aimed at particular load balancing topics which are more likely to have addressed other issues rather than load balancing problem are omitted. Hence, in the qualification of this review, it must be considered that this systematic review took into account studies published in the major international cloud computing journals.
- *Research questions:* The defined questions might not have covered the whole load balancing area, which implies the probability to define further relevant questions could be defined.
- *Study and Publication bias:* Five of the most reliable electronic databases were selected, relied on previous review experiences. In point of fact, the statistics indicate that this five electronic database would have to offer the most relevant and credible studies. However, selection of all applicable primary studies could not be guaranteed. There is a possibility that some applicable studies were overlooked throughout the processes mentioned in section 3. There could be various reasons, ranging from the search string to the data extraction. As accurate as possible, we attempted to prevent this by complying the references in primary studies.

6. Open issues and future trends

This section offers major load balancing techniques issues that have not been comprehensively and thoroughly studied as yet as research directions in the future. Empathizing the fundamental performance metrics of load and allocated resources and application evolvement of these metrics over time is indispensable to make practicable decisions in regards to resource reallocating/releasing. By addressing these issues, it must distinguish that empathizing the performance of any complex computing system such as cloud is intrinsically hard. The first reason is that cloud systems that are driven by computer operating systems do not offer real-time guarantees. Second and more importantly, to predict and control the performance of programs, there is a necessity for a foundational theory to direct us in constructing practical tools. This is a general case for computing systems, but due to the additional layer of virtualization on top of which cloud applications execute, it becomes, even more, prominent for cloud environments.

Furthermore, by discussing and analyzing the mentioned state of the art techniques, it has been observed that there is not any independent technique which addresses all issues involved in load balancing. For example, some techniques considered the QoS, reliability and scalability while some totally ignore these issues. Also, most of the reviewed mechanisms have been utilized with simulation to evaluate the proposed mechanisms, some others utilized none. Therefore, another fascinating point of future study would be to investigate the effects the size would have on systems performance in a large-scale environment by real cloud systems or utilizing some simulator such as CloudSim. As in this paper reviewed, research is underway towards decentralizing load balancing. In essence, this would appropriate, to treat a data center as a single pool of resources. However, it could be undesirable, in any failure case which may impact the operations of the entire system. Hence, an adaptive load balancing mechanism would be a more beneficial solution, clusters in which are dynamically formed, based on current request and application status, and resources in which are managed within clusters independently. Adaptive load balancing would empower controlling the trade-off between robust operation and efficient resource utilization and would probably involve a compounding of centralized and distributed control mechanism.

Additionally, in the most of the reviewed techniques, some factors such as energy consumption, carbon emission, and cost of service were not considered. These factors are discussed and analyzed independently; also, they are ready to be adopted to increase current balancing techniques popularity and effectiveness. The economy of scale is the benefits that advocate the adoption of the cloud. Energy saving is an essential factor to provide economic efficiency where maximized utilization of resources resulted from reduced set of resources. Moreover, in some cases, the task or VM migration processes can be improved to search the resources with lower cost. Therefore, new techniques which enable load balancing based on energy consumption, carbon emission and cost of service is extremely promising.

Another important concept for future research and works which get low attention in current load balancing mechanisms are Failure management and task migration features; therefore, they would have added to existing method for improving their efficiency. Furthermore, elasticity is a key feature of cloud computing, resources could be allocated or released automatically. Thus, how resource could be utilized or released, by maintaining the same performance as traditional systems. Additionally, resolving when to migrate and measure the load still has crucial research applications. With incomplete information, determining when to migrate represents the common issue of making decisions in a cloud environment. Examining prior requirements and current process properties, make possible to determine future load. Thus, research with a focus on predictive measures of load would be desirable. Importantly, for a virtualized infrastructure, thoroughly investigation of the barter between the efficiency utilizing of the essential hardware infrastructure and predictability of resources is a necessity. Heterogeneous load migration is majorly interesting research area, but it must be proven that it is practicable and cost-effective.

7. Summary and conclusion

This study presented a systematic review of load balancing techniques in cloud environments. In a similar way, we reviewed several state-of-the-art load balancing in cloud computing system, clarifying and discussing open issues via an in-depth analysis of over 15 primary studies among the basic 726 papers from our search query. Through answers provoked by three exploratory research questions, we found evidence confirming load balancing as an ascending mechanism that introduces a new paradigm by increasing cloud performance and impacting on resource utilization, also, ensures that each input request is distributed efficiently and fairly. Based on the available literature, we decided to classify the field into two subdomains related to dynamic load balancing and hybrid load balancing studies. We also discussed the advantages and disadvantages associated with several load balancing algorithms. The challenges of these algorithms are addressed so that more efficient load balancing techniques can be developed in future. Proper load balancing have the ability to keep minimum resource consumption which will conclude further reduction of energy consumption and carbon emission rate. In general, load balancing mechanisms in computing environment still need improvements in terms of managing the heterogeneity of its environment in order to become a truly on-demand technique, reducing associated overhead, service response time and improving performance etc. The overall data collected in this study help to acquaint the researchers with the state-of-the-art in the load balancing area. exclusively, the answers to the defined questions summarized load balancing's primary purpose, current challenges, open issues, approaches and mechanisms in cloud systems. We sincerely hope that the outcomes of this study will help researchers to develop new research fronts that further contribute to the maturity and adoption of load balancing in cloud computing.

References

Abdulhamid, S. i. M., et al. (2014). "Scheduling techniques in on-demand grid as a service cloud: A review." Journal of Theoretical and Applied Information Technology **63**: 10--19.

Abdulhamid, S. i. M. and M. S. A. Latiff (2014). "Tasks scheduling technique using League Championship Algorithm for makespan minimization in IaaS cloud." ARPN Journal of Engineering and Applied Sciences **9**: 2528--2533.

Abdullahi, M., et al. (2016). "Symbiotic Organism Search optimization based task scheduling in cloud computing environment." Future Generation Computer Systems **56**: 640-650.

Abrishami, S. and M. Naghibzadeh (2012). "Deadline-constrained workflow scheduling in software as a service Cloud." Scientia Iranica **19**(3): 680-689.

Ali M. Alakeel (2010). "A guide to dynamic load balancing in distributed computer systems." International Journal of Computer Science and Information Security: 153--160.

Arab, M. N. and M. Sharifi (2014). "A model for communication between resource discovery and load balancing units in computing environments." The Journal of Supercomputing **68**(3): 1538-1555.

Asghari, S. and N. J. Navimipour (2016). "Review and Comparison of Meta-Heuristic Algorithms for Service Composition in Cloud Computing." Majlesi Journal of Multimedia Processing **4**(4).

Ashouraie, M., et al. (2015). "Priority-based task scheduling on heterogeneous resources in the Expert Cloud." Kybernetes **44**(10).

Banerjee, S., et al. (2015). "Development and Analysis of a New Cloudlet Allocation Strategy for QoS Improvement in Cloud." Arabian Journal for Science and Engineering **40**(5): 1409-1425.

Beloglazov, A., et al. (2012). "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing." Future Generation Computer Systems **28**(5): 755-768.

Boettcher, S. and A. G. Percus (1999). "Extremal optimization: methods derived from coevolution." Proceedings of the Genetic and Evolutionary Computation Conference (GECCO99): 825-832.

Buyya, R., et al. (2011). Cloud Computing Principles and Paradigms, Wiley Publishing.

Celesti, A., et al. (2012). "Virtual machine provisioning through satellite communications in federated Cloud environments." Future Generation Computer Systems **28**(1): 85-93.

Chang, V. "An overview, examples and impacts offered by Emerging Services and Analytics in Cloud Computing." International Journal of Information Management.

Charband, Y. and N. J. Navimipour (2016). "Online knowledge sharing mechanisms: a systematic review of the state of the art literature and recommendations for future research." Information Systems Frontiers: 1-21.

Chen, S.-L., et al. (2016). "CLB: A novel load balancing architecture and algorithm for cloud services." Computers & Electrical Engineering.

Chiregi, M. and N. Jafari Navimipour (2016). "Trusted services identification in the cloud environment using the topological metrics." Karbala International Journal of Modern Science.

Chiregi, M. and N. J. Navimipour (2016). "A new method for trust and reputation evaluation in the cloud environments using the recommendations of opinion leaders' entities and removing the effect of troll entities." Computers in Human Behavior **60**: 280-292.

Cho, K.-M., et al. (2014). "A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing." Neural Computing and Applications.

Chou, D. C. (2015). "Cloud computing: A value creation model." Computer Standards & Interfaces **38**: 72-77.

Daraghmi, E. Y. and S.-M. Yuan (2015). "A small world based overlay network for improving dynamic load-balancing." Journal of Systems and Software **107**: 187-203.

De Falco, I., et al. (2015). "Extremal Optimization applied to load balancing in execution of distributed programs." Applied Soft Computing **30**: 501-513.

Dodonov, E. and R. F. de Mello (2010). "A novel approach for distributed application scheduling based on prediction of communication events." Future Generation Computer Systems **26**(5): 740-752.

F. Liu, J. T., J. Mao, R. Bohn, J. Messina, L. Badger and D. Leaf (2011). "NIST Cloud Computing Reference Architecture." NIST Special Publication **500-292**: 35.

Fang, Y., et al. (2010). A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing. Web Information Systems and Mining: International Conference, WISM 2010, Sanya, China, October 23-24, 2010. Proceedings. F. L. Wang, Z. Gong, X. Luo and J. Lei. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 271-277.

Gutierrez-Garcia, J. O. and A. Ramirez-Nafarrate (2015). "Agent-based load balancing in Cloud data centers." Cluster Computing.

Jafari Navimipour, N. (2011). "Control the Topology and Increase the Tolerance of Heterogeneous Wireless Sensor." International Journal of Advanced Research in Computer Science **2**(6).

Jafari Navimipour, N. and Y. Charband (2016). "Knowledge sharing mechanisms and techniques in project teams: literature review, classification, and current trends." Computers in Human Behavior.

Jafari Navimipour, N. and S. H. Es-Hagi (2011). "Reduce Energy Consumption and Increase the Lifetime of Heterogeneous Wireless Sensor Networks: Evolutionary Approach." International Journal of Advanced Research in Computer Science **2**(5).

Jafari Navimipour, N., et al. (2014). "Job scheduling in the Expert Cloud based on genetic algorithms." Kybernetes **43**(8): 1262-1275.

Jafari Navimipour, N., et al. (2015). "Expert Cloud: A Cloud-based framework to share the knowledge and skills of human resources." Computers in Human Behavior **46**(C): 57-74.

Jose Moura, D. H. (2015). "Review and Analysis of Networking Challenges in Cloud Computing." Journal of Network and Computer Applications.

Kalra, M. and S. Singh (2015). "A review of metaheuristic scheduling techniques in cloud computing." Egyptian Informatics Journal **16**(3): 275-295.

Kanakala, V. R., et al. (2015). Performance analysis of load balancing techniques in cloud computing environment. Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference on.

Katyal, M. and A. Mishra (2013). "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment." International Journal of Distributed and Cloud Computing **1**(2).

Khanli, L. M. and S. N. Razavi (2008). LGR: The New Genetic Based Scheduler for Grid Computing Systems. Computational Intelligence for Modelling Control & Automation, 2008 International Conference on, IEEE.

Kitchenham, B. (2004). "Procedures for Performing Systematic Reviews." Keele University Technical Report.

Kitchenham, B., et al. (2009). "Systematic literature reviews in software engineering – A systematic literature review." Information and Software Technology **51**(1): 7-15.

Kupiainen, E., et al. (2015). "Using metrics in Agile and Lean Software Development – A systematic literature review of industrial studies." Information and Software Technology **62**: 143-163.

L.D, D. B. and P. Venkata Krishna (2013). "Honey bee behavior inspired load balancing of tasks in cloud computing environments." Applied Soft Computing **13**(5): 2292-2303.

Lai, K.-C. and Y.-F. Yu (2011). "A scalable multi-attribute hybrid overlay for range queries on the cloud." Information Systems Frontiers **14**(4): 895-908.

Li, K. (2012). "Optimal Load Distribution for Multiple Heterogeneous Blade Servers in a Cloud Computing Environment." Journal of Grid Computing **11**(1): 27-46.

Liu, Y., et al. (2015). "DeMS: A hybrid scheme of task scheduling and load balancing in computing clusters." Journal of Network and Computer Applications.

Lu, Y., et al. (2011). "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services." Performance Evaluation **68**(11): 1056-1071.

Madni, S. H. H., et al. (2016). An Appraisal of Meta-Heuristic Resource Allocation Techniques for IaaS Cloud.

Maguluri, S. T., et al. (2012). Stochastic models of load balancing and scheduling in cloud computing clusters. INFOCOM, 2012 Proceedings IEEE.

Marston, S., et al. (2011). "Cloud computing — The business perspective." Decision Support Systems **51**(1): 176-189.

Milani, B. A. and N. J. Navimipour (2016). "A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions." Journal of Network and Computer Applications **64**: 229-238.

Mohamed, N., et al. (2013). "A dual-direction technique for fast file downloads with dynamic load balancing in the Cloud." Journal of Network and Computer Applications **36**(4): 1116-1130.

Mohamed, N., et al. (2013). "DDOps: dual-direction operations for load balancing on non-dedicated heterogeneous distributed systems." Cluster Computing **17**(2): 503-528.

Mohammad Aghdam, S. and N. Jafari Navimipour (2016). "Opinion leaders selection in the social networks based on trust relationships propagation." Karbala International Journal of Modern Science.

Naha, R. K. and M. Othman (2014). "Brokering and Load-Balancing Mechanism in the Cloud – Revisited." IETE Technical Review **31**(4): 271-276.

Nakai, A., et al. (2014). "On the Use of Resource Reservation for Web Services Load Balancing." Journal of Network and Systems Management **23**(3): 502-538.

Navimipour, N. J. (2011). "Control the Topology and Increase the Tolerance of Heterogeneous Wireless Sensor Networks." International Journal of Advanced Research in Computer Science **2**(6).

Navimipour, N. J. (2015). "A formal approach for the specification and verification of a Trustworthy Human Resource Discovery mechanism in the Expert Cloud." Expert Systems with Applications **42**(15): 6112-6131.

Navimipour, N. J. (2015). "Task scheduling in the Cloud Environments based on an Artificial Bee Colony Algorithm."

Navimipour, N. J. and L. M. Khanli (2008). The LGR method for task scheduling in computational grid. Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on, IEEE.

Navimipour, N. J. and N. Khezr (2015). "MapReduce and its application in optimization algorithms: A comprehensive study." Majlesi Journal of Multimedia Processing **4**(3).

Navimipour, N. J. and F. S. Milani (2014). "A comprehensive study of the resource discovery techniques in Peer-to-Peer networks." Peer-to-Peer Networking and Applications **8**(3): 474-492.

Navimipour, N. J. and F. S. Milani (2015). "A comprehensive study of the resource discovery techniques in Peer-to-Peer networks." Peer-to-Peer Networking and Applications **8**(3): 474-492.

Navimipour, N. J. and F. S. Milani (2015). "Task scheduling in the cloud computing based on the cuckoo search algorithm." International Journal of Modeling and Optimization **5**(1): 44.

Navimipour, N. J., et al. (2015). "Behavioral modeling and automated verification of a Cloud-based framework to share the knowledge and skills of human resources." Computers in Industry **68**: 65-77.

Navimipour, N. J. and A. M. Rahmani (2009). The New Genetic Based Method with Optimum Number of Super Node in Heterogeneous Wireless Sensor Network for Fault Tolerant System. Intelligent Networking and Collaborative Systems, 2009. INCOS'09. International Conference on, IEEE.

Navimipour, N. J., et al. (2014). "Resource discovery mechanisms in grid systems: A survey." Journal of Network and Computer Applications **41**: 389-410.

Navimipour, N. J., et al. (2015). "Expert Cloud: A Cloud-based framework to share the knowledge and skills of human resources." Computers in Human Behavior **46**: 57-74.

Navimipour, N. J., et al. (2012). "Minimize Energy Consumption and Improve the Lifetime of Heterogeneous Wireless Sensor Networks by Using Monkey Search Algorithm." 2012 International Conference on Information and Knowledge Management, IPCSIT **45**: 42-47.

Navimipour, N. J. and Z. Soltani (2016). "The impact of cost, technology acceptance and employees' satisfaction on the effectiveness of the electronic customer relationship management systems." Computers in Human Behavior **55**: 1052-1066.

Navimipour, N. J. and B. Zareie (2015). "A model for assessing the impact of e-learning systems on employees' satisfaction." Computers in Human Behavior **53**: 475-485.

Navin, A. H., et al. (2014). "Expert grid: new type of grid to manage the human resources and study the effectiveness of its task scheduler." Arabian Journal for Science and Engineering **39**(8): 6175-6188.

Niu, J., et al. (2008). Characterizing effective auction mechanisms: insights from the 2007 TAC market design competition. Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 2. Estoril, Portugal, International Foundation for Autonomous Agents and Multiagent Systems**:** 1079-1086.

Ramezani, F., et al. (2013). "Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization." International Journal of Parallel Programming **42**(5): 739-754.

Rathore, N. and I. Chana (2014). "Load Balancing and Job Migration Techniques in Grid: A Survey of Recent Trends." Wireless Personal Communications **79**(3): 2089-2125.

Remesh Babu, K. R. and P. Samuel (2016). Enhanced Bee Colony Algorithm for Efficient Load Balancing and Scheduling in Cloud. Innovations in Bio-Inspired Computing and Applications: Proceedings of the 6th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2015) held in Kochi, India during December 16-18, 2015. V. Snášel, A. Abraham, P. Krömer, M. Pant and K. A. Muda. Cham, Springer International Publishing**:** 67-78.

Sharif, S. H., et al. (2013). "A review on search and discovery mechanisms in social networks." International Journal of Information Engineering and Electronic Business **5**(6): 64.

Soltani, Z. and N. J. Navimipour (2016). "Customer relationship management mechanisms: A systematic review of the state of the art literature and recommendations for future research." Computers in Human Behavior **61**: 667-688.

Souri, A. and N. J. Navimipour (2014). "Behavioral modeling and formal verification of a resource discovery approach in Grid computing." Expert Systems with Applications **41**(8): 3831-3849.

Voorsluys, W., et al. (2011). Introduction to Cloud Computing. Cloud Computing, John Wiley & Sons, Inc.**:** 1-41.

Voorsluys, W., et al. (2009). Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation. Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings. M. G. Jaatun, G. Zhao and C. Rong. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 254-265.

Wang, T., et al. (2012). "MBA: A market-based approach to data allocation and dynamic migration for cloud database." Science China Information Sciences **55**(9): 1935-1948.

Wang, Z., et al. (2015). "Workload balancing and adaptive resource management for the swift storage system on cloud." Future Generation Computer Systems **51**: 120-131.

Wickremasinghe, B., et al. (2010). CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications. 2010 24th IEEE International Conference on Advanced Information Networking and Applications.

Xu, G., et al. (2013). "A load balancing model based on cloud partitioning for the public cloud." Tsinghua Science and Technology **18**(1): 34-39.

Zareie, B. and N. Jafari Navimipour (2016). "The Effect of Electronic Learning Systems on the Employee's Commitment." The International Journal of Management Education.

Zhang, Z. and X. Zhang (2010). A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation. Industrial Mechatronics and Automation (ICIMA), 2010 2nd International Conference on.

Highlights

- providing an overview of existing challenges in a range of problem domains associated with cloud computing that can be addressed using load balancing
- providing a systematic overview of the existing techniques for load balancing, and the manner in which these have been applied to cloud computing
- exploring the future challenges for cloud computing and the role that load balancing can play
- outlining the key areas where future research can improve the use of load balancing techniques in cloud computing