第 49 卷 第 7 期
2022 年 7 月

湖南大学学报（自然科学版）
Journal of Hunan University (Natural Sciences）

Vol. 49 No. 7
July 2022

Open Access Article https://doi.org/10.55463/issn.1674-2974.49.7.5

Modern Load Balancing Techniques and Their Effects on Cloud Computing

Asan Baker Kanbar[1,2*], Kamaran Faraj[3]

[1]Technical Collage of Informatics, Sulaimani Polytechnic University ,Slaimani 46001, Kurdistan Region, Iraq,
[2]Department of Computer Science, Cihan University Sulaimaniya, Sulaimaniya 46001, Kurdistan Region, Iraq,
[3]Department of Computer Science, Sulaimaniyah University, Slemani, Kurdistan Region, Iraq,

Abstract: Cloud computing is a new and advanced viewpoint for large-scale parallel and distributed computing systems. Cloud computing is growing quickly, and users are demanding more services and better results, so cloud-computing load balancing has become a very thought-provoking and important research area. Load on the cloud is growing extremely with the expansion of new applications. Load balancing is a major area of the cloud computing environment, which guarantees that all connected devices or processors simultaneously perform the same amount of work. Hence, an efficient load-balancing scheme is needed to improve the performance of cloud computing. Different researchers in the past years have proposed several load-balancing algorithms. This paper examines the important necessities and concerns for designing and implementing a suitable load balancer for cloud environments. In addition, we constitute an entire survey of recently proposed cloud load balancing solutions; finally, we propose evaluating these solutions based on suitable metrics and discuss their advantages and disadvantages.
Keywords: cloud computing, load balancing, task, scheduling, resource allocation.

现代负载平衡技术及其对云计算的影响

摘要：云计算是大规模并行和分布式计算系统的一个新的和先进的观点。云计算发展迅速，用户要求更多的服务和更好的结果，因此云计算负载均衡成为一个非常值得思考和重要的研究领域。随着新应用程序的扩展，云上的负载正在急剧增长。负载平衡是云计算环境的一个主要领域，它保证所有连接的设备或处理器同时执行相同数量的工作。因此，需要一种有效的负载平衡方案来提高云计算的性能。在过去的几年中，不同的研究人员提出了几种负载平衡算法。本文探讨了为云环境设计和实施合适的负载均衡器的重要必要性和关注点。此外，我们对最近提出的云负载平衡解决方案进行了全面调查；最后，我们建议根据合适的指标评估这些解决方案，并讨论它们的优缺点。

关键词：云计算、负载均衡、任务、调度、资源分配。

1. Introduction

Cloud Computing has become very popular in recent times. Cloud computing refers to maintaining a collection of diverse computer resource systems and services that can be easily configured and provided to users over the internet, with minimal efforts. Cloud computing provides various benefits such as speed, security, and privacy [1, 18]. An enormous number of users may need to access the cloud at the same time; this may cause applications running concurrently to face many challenges in balancing the load amongst them. The entire system can even break because of

Received: April 19, 2022 / Revised: May 11, 2022 / Accepted: June 13, 2022 / Published: July 30, 2022

About the authors: Asan Baker Kanbar, Technical Collage of Informatics, Sulaimani Polytechnic University, Slemani, Iraq; Department of Computer Science, Cihan University Sulaimaniya, Sulaimaniya, Iraq; Kamaran Faraj , Department of Computer Science, Sulaimaniyah University, Slemani, Iraq

Corresponding author Asan Baker Kanbar, asan.baker@spu.edu.iq, asan.baker@sulicihan.edu.krd

overloading. Load balancing is a method that is used to ensure that all the computing resources are distributed efficiently. The resources in cloud computing are offered to users on demand, as per SLA requirements. For each application service in the data center, providers should assure Quality of Service (QoS) while achieving optimal utilization of the server and energy [2]. In the cloud, dynamic resources can be efficiently managed using virtualization technologies (VM resources) that balance the workload for the entire system, and schedule and allocate the resources efficiently [3]. In this paper, we will explain the types of cloud computing and compare and discuss the advantages and disadvantages of cloud computing algorithms based on speed, replication, single point of failure (SPOF), spatial distribution, heterogeneity, network overhead, implementation complexity, and fault tolerance.

2. Layers and Types of Cloud

Clouds are classified based on the capabilities and services provided at the abstraction level. In the architecture illustrated in Fig. 1, higher layer services can be constituted from those at the lower layers.

2.1. IaaS - Infrastructure as a Service

IaaS provides on-demand access to fundamental computing resources, virtualized resources, storage, computation, communication, network hardware, and associated software. Microsoft Azure, Google Compute Engine, Cisco Metapod, and Amazon Web Services are the major providers of IaaS. In the IaaS called Hardware as a Service (HaaS), platform virtualization is done in IaaS.

2.2. PaaS-Platform as a Service

PaaS is an application development and deployment platform. With it, the cloud provider hosts everything like servers, networks, storage, operating system software, middleware, and databases at their data center. Developers can create and deploy applications without knowing how many processors or how much data those applications will need. Just the internet connection is needed to develop and host whatever is needed on a single cloud platform.

2.3. SAAS-Software as a Service

Software as a Service (SAAS), also known as cloud-based software or cloud applications, is application software that's hosted in the cloud. A web portal is required to access services at this layer, as is a web browser and a dedicated desktop client, or an API that integrates with your desktop or mobile operating system. There is no need for locally installed programs. Traditional desktop applications like MS Word and Spreadsheet can now be accessed online. Also, with SAAS programs there is no need for software maintenance: users can simply develop and test the applications. An example of SAAS is Salesforce.com: it provides CRM applications that completely reside on their servers.

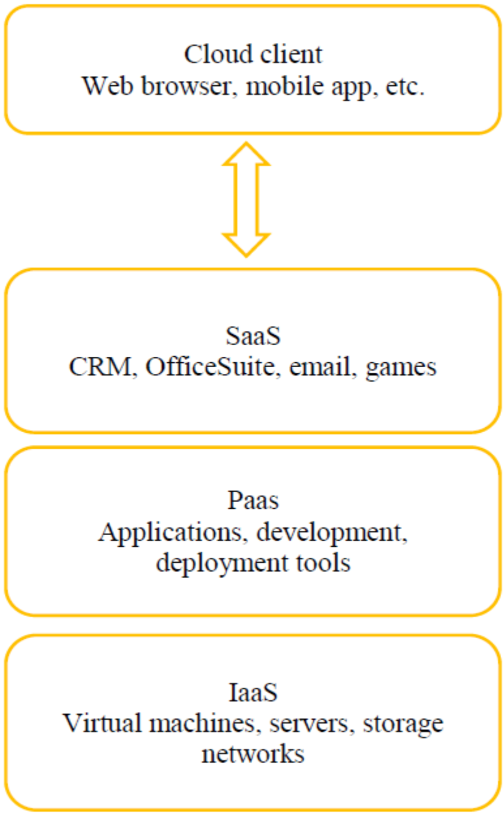

Fig. 1 Service models in cloud computing

3. Deployment Models in Cloud Computing

There are four types of deployment models in cloud computing; their type is determined based on their physical location and distribution. The deployment model types are illustrated in Fig. 2.

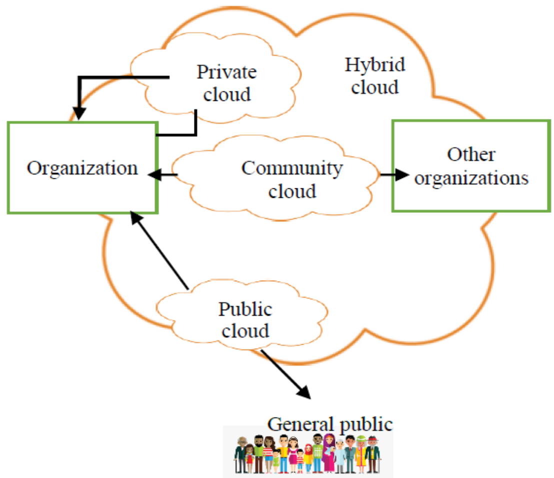

Fig. 2 Cloud computing deployment models

3.1. Public Cloud

A public cloud is a type of cloud hosting that provides easy access to its users. Public clouds are maintained by a third party. All the services are available to the general public on a pay-as-you-go basis.

3.2. Private Cloud

A private cloud is also named an 'internal cloud', which permits the accessibility of systems and services within a limited boundary or organization. The cloud platform is implemented in a cloud-based secure environment that is protected by advanced firewalls under the observation of a specific organization's IT department. It is a cloud model that is used by a company for its internal/partner use, providing the organizations with greater control over data and its security. Business organizations that have dynamic, critical, secured, management demand-based requirements should use a private cloud.

3.3. Hybrid Cloud

A hybrid cloud is another cloud computing type, which is a deployment model that combines the public and private clouds; i.e., it can be a combination of two or more cloud servers, i.e., private and public [17]. It can combine these under one architecture, or the clouds can remain as separate entities. Non-critical tasks like workload development and test are done using the public cloud. In contrast, critical tasks such as organization data handling are done using a private cloud.

3.4. Community Cloud

In the community cloud, setup is shared manually among organizations in the same community or area. This model is managed and hosted domestically or by a third-party vendor.

4. Related Works

Basic load balancing is the method of distributing requests to resources in the cloud computing environment to optimize results. There are many constraints on formerly proposed load balancing algorithms. The First Come First Serve (FCFS) is the simplest load balancing algorithm that considers only the time used by the task to reach the VM. The Round Robin (RR) load balancing algorithm is similar to FCFS with only one difference in which it gives the same time slots to all tasks. Research on previous load balancing algorithms, like Shortest Job First (SJF) and FCFS, shows that throughput was not improved much by these implementations [4]. [5] defined throughput as the number of requests served in a single time unit. With higher throughput, cloud computing systems will have higher performance. All cloud frameworks must adapt to any potential failure while easily maintaining the identical throughput. For example, if a node fails in a cloud framework, the framework cannot fail. A load-balancing algorithm can automatically reassign a request from a failing node to any other available node in the most feasible way. [6] proposed a distributed and scalable load-balancing mechanism for cloud computing using game theory. This mechanism is implemented locally in each PM, and its current state is then detected. Then, if under-used or over-used states are detected, a selection process begins. Through this process, the appropriate moving VMs are selected. Eventually, the best host for moving VMs is selected by a migration process. [7] introduced a load-balancing algorithm for resource allocation in cloud computing. This algorithm is based on the hybridization of teaching–learning-based optimization and grey wolf optimization algorithms to maximize the throughput. The algorithm uses a well-balanced load across VMs and solves the local optimum trap problem. Task priorities are also well balanced, and load-balancing is effective based on the minimization of task time in the queue. [8] introduced the min-min algorithm, which starts with a set of all pending jobs. First, the time required to finish an activity is calculated. Next, the work with the least completion time is selected. Finally, the chosen node and job are mapped. The node's ready time is then updated. This process is repeated until all jobs are assigned. This algorithm is executed within the minimum time but leads to starvation, and therefore is not suitable for a dynamic environment. Singh and Chana [9] introduced an automatic resource planning structure called fuzzy logic-based energy use programming. It plans cloud resources in a data center. This framework has been validated in the CloudSim-based simulation environment. The program, however, provides poorer CC service performance than other existing methods. [10] presented a new planning algorithm based on a GA for scheduling tasks. Scheduling of tasks and resource allocation are the two significant factors in CC. This model includes four attributes for planning purposes: time to complete the activity, activity expenses, reliability, and bandwidth. This model has additionally followed the crossover and the mutation operation rules to increase quality service. But GA did not function well if resources were strictly bound. For resource allocation, Wei et al. [11] have introduced the imperfect information game using the hidden Markov model (HMM) method. It is used for resource distribution, similar to the Stackelberg game. This model maximizes the revenue of the provider and resource requester. Initially, the current bid of the service provider is predicted by the HMM. To guarantee the ideal return to the infrastructure provider, the DRA model is proposed, which supports synchronous deployment for multiple service providers and resources. Pillai and Rao [12] proposed a mechanism for allocating resources in cloud technologies that depends on the coalition formation and insecurity principles of game theory. This technique avoids inter-programming complexities by solving the problem of optimizing coalition formation.

5. Load Balancing Based on the Cloud Environment

Cloud computing can have either a static or dynamic environment based on how the developer configures the cloud environment demanded by the cloud provider.

5.1. Static Environment

In a static environment, the cloud supplier installs homogeneous resources. However, the resources in the cloud are not flexible when the environment is static. The cloud requires prior knowledge of the node capacity, memory, processing power, performance, and statistics required by the users. These user requirements are not subject to alteration during runtime. Algorithms proposed to achieve load balancing in a static environment cannot adapt to runtime alterations in load. Although static environments are easier to simulate, they are not appropriate for heterogeneous cloud environments.

The Round Robin algorithm provides load balancing in a static environment [13]. In a static environment, the resources are supplied to the task on a FCFS basis (the task that entered first will be first allocated the resource) and scheduled through a time-sharing method. The resource that is the least loaded is allocated to the task. Eucalyptus uses greedy (first fit) with Round Robin for VM mapping. Radojevic [14] introduced and developed an algorithm over Round Robin known as the Central Load Balancing Decision Model (CLBDM). Although it uses Round Robin as its basis, it also measures the duration of the client–server connection by calculating the overall execution time of the task on the given cloud resource.

5.2. Dynamic Environment

In a dynamic cloud computing environment, the cloud provider installs heterogeneous resources. The resources are flexible in a dynamic environment. In this scenario, the cloud cannot depend on the former knowledge, whereas it takes into account runtime statistics. The user requirements are granted flexibility (i.e., they may change during runtime). Algorithms proposed to achieve load balancing in dynamic environments can easily adapt to runtime changes in load. A dynamic environment is hard to simulate but is extremely adaptable with a cloud computing environment. Based on weighted least connection (WLC) [15], Ren introduced a load-balancing technique in a dynamic environment called ESWLC. It assigns the resource with the least weight to a task and takes node capabilities into account. Taking the resource weights and node capabilities into consideration, a task is allocated to a node. The Load Balancing Min-Min algorithm suggested in [16] uses three-level frameworks for resource allocation in a dynamic cloud computing environment. It uses the opportunistic load balancing algorithm as its basis. As the cloud is enormously scalable and autonomous, dynamic scheduling is a better choice than static scheduling.

Table 1 Advantages and disadvantages of load balancing algorithms

	Pros	**Cons**
CLDBM	• Solves issues of Round Robin Algorithm • Automated tasks forwarding reduces the need for a human administrator	• Inherits Round Robin issues such as not taking into consideration node Capabilities • Single point of failure (if CLBDM fails, the whole process fails) • The threshold might not be applied to all cases
INS	Initially proved to handle some sort of dynamic load balancing	• No forecasting algorithm to identify the future behavior of the nodes • Complicated in terms of implementation • Only certain parameters are considered, such as distance and time
ESWLC	• More accurate results than WLC	• Complicated • The prediction algorithm requires existing data and has a long processing time
Enhanced MapReduce	• Less overhead for the reduce tasks	• High processing time • Reduce tasks capabilities are not taken into consideration
Ant Colony	• Best case scenario is that the underloaded node is found at the beginning of the search • Decentralized, no single point of failure • Ants can collect the information faster	• Network overhead because of the large number of ants • Points of initiation of ants and number of ants are not clear • Node status change after ants' visits to them is not taken into account • Only availability of node is being considered, while there are other factors that should be taken into consideration
VM Mapping	• Reliable calculation method	• Single Point of failure • Does not take into account network load and node capabilities
DDFTP	• Fast • Reliable download of files	• Full replication of data files that requires high storage in all nodes

6. Comparison and Discussion

Load balancing is an essential task in the cloud computing environment to achieve maximum resource utilization. In the current study, we surveyed many algorithms and compared them based on speed, replication, SPOF, spatial distribution, heterogeneity, network overhead, implementation complexity, and fault tolerance. CLDBM can handle heterogeneous resources, offer efficient task distribution, and provide minimum node idle time but cannot tolerate high delays. INS has moderate speed and full replication and can handle heterogeneous resources, offer efficient task distribution, and provide minimum node idle time but also cannot tolerate high delays. ESWLC is very fast, reduces query latency, avoids the problem of failure at overloaded nodes, can handle heterogeneous resources, offers efficient task distribution, and can tolerate high delays but has high implementation complexity. The MapReduce algorithm is slow but has no SPOF, reduces query latency, avoids the problem of failure at overloaded nodes, can handle heterogeneous resources and tolerate high delays, and offers efficient task distribution. However, it has high implementation complexity. The ant colony optimization algorithm can copy directory data automatically and quickly and can change from one directory server to another, but it is inefficient in task distribution, cannot handle heterogeneity resources, and has high fault tolerance. VM mapping is inefficient in task distribution and has high implementation complexity but can handle heterogeneity and has fault tolerance. The current design of DDFTP enables it to tolerate high delays, handle heterogeneous resources, efficiently adjust to dynamic operational conditions, offer efficient task distribution, and provide minimum node idle time, but it relies on full replication of the files on multiple sites, which wastes storage resources. Table 2 summarizes the study results.

Table 2 Comparison of load balancing algorithms

Algorithms	**Speed**	**Replication**	**SPOF**	**Task distribution**	**Heterogeneity**	**Network Overhead**	**Implementation Complexity**	**Fault Tolerance**
CLDBM	SLOW	Full	Yes	Yes	Yes	Yes	Low	No
INS	Moderate	Partial	Yes	Yes	Yes	Yes	High	No
ESWLC	FAST	Full	No	Yes	Yes	Yes	High	Yes
Enhanced MapReduce	SLOW	Full	No	Yes	Yes	Yes	High	Yes
Ant Colony	FAST	Full	No	No	No	Yes	Low	Yes
VM Mapping	FAST	Full	Yes	No	Yes	Yes	High	Yes
DDFTP	FAST	Full	No	Yes	Yes	No	Low	Yes

7. Conclusion

In this paper, we discuss various load-balancing techniques, each of which has advantages and disadvantages. We also discuss the challenges that must be overcome to provide the most suitable and efficient load-balancing algorithms. The static load-balancing technique affords the easiest simulation and monitoring of the environment but fails to model the heterogeneous nature of the cloud. Dynamic load-balancing algorithms, on the other hand, are hard to simulate but are well suited in heterogeneous cloud computing environments. In addition, the node level that implements the static or dynamic algorithm plays a significant role in determining the efficiency of the algorithm. In contrast to the centralized algorithm, the distributed algorithm offers better fault tolerance but requires a higher degree of replication. On the other hand, the hierarchical algorithm divides the load at different hierarchy levels, with the upper-level nodes requesting the services of the lower-level nodes in a balanced manner. Consequently, dynamic load-balancing techniques in distributed or hierarchical environments provide better performance. Nevertheless, the performance of the cloud computing environment can be further maximized if the dependencies between tasks are modeled using workflows. Additionally, according to our study outcomes, the high-quality load-balancing approaches in cloud computing are those that have dynamic, distributed, and non-cooperative features. Dynamic characteristics refer to dynamic and on-demand properties in the cloud that result in dynamic workloads and services. Distributed designing can prevent bottlenecks and SPOFs and therefore provides better scalability and fault tolerance. Finally, the non-cooperative method can avoid unnecessary overheads and has a functioning load balancer.

This work may encourage beginners in the field to design novel techniques that may consider almost all metrics. In addition, some other important parameters can be taken into account, such as carbon emissions, energy consumption, and intelligence for green and intelligent load balancers for cloud computing environments.

References

[1] ALWORAFI M. A., DHARI A., AL-HASHMI A. A., DAREM A. B., and SURESHA. An improved SJF scheduling algorithm in cloud computing environment. Proceedings of the International Conference on Electrical,

Electronics, Communication, Computer and Optimization Techniques, Mysuru, 2016, pp. 208-212. https://doi.org/10.1109/ICEECCOT.2016.7955216

[2] UPADHYAYA J., & AHUJA N. J. Quality of service in cloud computing in higher education: A critical survey and innovative model. Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), Palladam, 2017, pp. 137–140. https://doi.org/10.1109/I-SMAC.2017.8058324

[3] KATYAL M., & MISHRA A. A comparative study of load balancing algorithms in cloud computing environment. *International Journal of Distributed and Cloud Computing*, 2013, 1(2): 5-14. https://www.i-scholar.in/index.php/ijdcc/article/view/50722

[4] ABED S., & SHUBAIR D. S. Enhancement of task scheduling technique of big data cloud computing. Proceedings of the International Conference on Advances in Big Data, Computing and Data Communication Systems, Durban, 2018, pp. 1-6. https://doi.org/10.1109/ICABCD.2018.8465422

[5] KANERIA O., & BANYAL R. Analysis and improvement of load balancing in cloud computing. Proceedings of the International Conference on ICT in Business Industry & Government, Indore, 2016, pp. 1-5. https://doi.org/10.1109/ICTBIG.2016.7892711

[6] KHANI H., YAZDANI N., and MOHAMMADI S. A self-organized load-balancing mechanism for cloud computing. *Concurrency and Computation: Practice and Experience*, 2017, 29(4): e3897. https://doi.org/10.1002/cpe.3897

[7] MOUSAVI S., MOSAVI A., and VARKONYI-KOCZY A. R. A load-balancing algorithm for resource allocation in cloud computing. In: LUCA D., SIRGHI L., and COSTIN C. (eds.) *Recent Advances in Technology Research and Education. INTER-ACADEMIA 2017. Advances in Intelligent Systems and Computing*, Vol. 660. Springer, Cham, 2018: 289-296. https://doi.org/10.1007/978-3-319-67459-9_36

[8] CHEN J., LI K., TANG Z., BILAL K., YU S., WENG C., and LI K. A parallel random forest algorithm for big data in a spark cloud computing environment. *IEEE Transactions on Parallel and Distributed Systems*, 2017, 28(4): 919-933. https://doi.org/10.1109/TPDS.2016.2603511

[9] SINGH S., & CHANA I. EARTH: energy-aware autonomic resource scheduling in cloud computing. *Journal of Intelligent & Fuzzy Systems*, 2016, 30(3): 1581–1600. https://doi.org/10.3233/IFS-151866

[10] MA J., LI W., FU T., YAN L., and HU G. A novel dynamic task scheduling algorithm based on improved genetic algorithm in cloud computing. In: ZENG Q. A. (eds.) *Wireless Communications, Networking and Applications. Lecture Notes in Electrical Engineering*, Vol. 348. Springer, New Delhi, 2016: 829–835. https://doi.org/10.1007/978-81-322-2580-5_75

[11] WEI W., FAN X., SONG H., FAN X., and YANG J. Imperfect information dynamic stackelberg game based resource allocation using hidden Markov for cloud computing. *IEEE Transactions on Services Computing*, 2018, 11(1): 78–89. https://doi.org/10.1109/TSC.2016.2528246

[12] PILLAI P. S., & RAO S. Resource allocation in cloud computing using the uncertainty principle of game theory. *IEEE Systems Journal*, 2016, 10(2): 637–648. https://doi.org/10.1109/JSYST.2014.2314861

[13] SOTOMAYOR B., MONTERO R. S., LLORENTE I. M., and FOSTER I. Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing*, 2009, 13(5): 14-22. https://doi.org/10.1109/MIC.2009.119

[14] RADOJEVIC B., & ZAGAR M. Analysis of issues with load balancing algorithms in hosted (cloud) environments. Proceedings of the 34th International Convention MIPRO, Opatija, 2011, pp. 416-420. https://ieeexplore.ieee.org/document/5967092

[15] LEE R., & JENG B. Load-balancing tactics in cloud. Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Beijing, 2011, pp. 447-454. https://doi.org/10.1109/CyberC.2011.79

[16] WANG S. C., YAN K. Q., LIAO W. P., and WANG S. S. Towards a load balancing in a three-level cloud computing network. Proceedings of the 3rd International Conference on Computer Science and Information Technology, Chengdu, 2010, pp. 108-113. https://doi.org/10.1109/ICCSIT.2010.5563889

[17] KADIM U. N., & MOHAMMED I. J. A Hybrid Software Defined Networking-Based Load Balancing and Scheduling Mechanism for Cloud Data Centers. *Journal of Southwest Jiaotong University*, 2020, 55(3). https://doi.org/10.35741/issn.0258-2724.55.3.3

[18] ABDUL-JABBAR S. S., ALDUJAILI A., MOHAMMED S. G., and SAEED H. S. Integrity and Security in Cloud Computing Environment: A Review. *Journal of Southwest Jiaotong University*, 2020, 55(1). https://doi.org/10.35741/issn.0258-2724.55.1.11

参考文:

[1] ALWORAFI M.A.、DHARI A.、AL-HASHMI A.A.、DAREM A.B. 和 SURESHA。云计算环境下改进的SJF调度算法电气、电子、通信、计算机和优化技术国际会议论文集，迈索尔，2016年，第 208-212 页。https://doi.org/10.1109/ICEECCOT.2016.7955216

[2] UPADHYAYA J., & AHUJA N. J. 高等教育中云计算的服务质量：一项重要的调查和创新模型。我-SMAC（社交、移动、分析和云中的物联网）国际会议论文集，帕拉丹，2017年，第 137-140 页。https://doi.org/10.1109/I-SMAC.2017.8058324

[3] KATYAL M., & MISHRA A. 云计算环境中负载平衡算法的比较研究。国际分布式与云计算杂志，2013, 1(2): 5-14. https://www.i-scholar.in/index.php/ijdcc/article/view/50722

[4] ABED S., & SHUBAIR D. S. 增强大数据云计算任务调度技术。大数据、计算和数据通信系统进展国际会议论文集，德班，2018年，第 1-6 页。https://doi.org/10.1109/ICABCD.2018.8465422

[5] KANERIA O., & BANYAL R. 云计算负载均衡分析与改进。商业和政府信息通信技术国际会议论文集，印多尔，2016年，第 1-5 页。https://doi.org/10.1109/ICTBIG.2016.7892711

[6] KHANI H.、YAZDANI N. 和 MOHAMMADI S. 云计算的自组织负载平衡机制。并发与计算：实践与经

验，2017，29(4)：e3897。https://doi.org/10.1002/cpe.3897
[7] MOUSAVI S.、MOSAVI A. 和 VARKONYI-KOCZY A. R. 云计算资源分配的负载平衡算法。在：LUCA D.、SIRGHI L. 和 COSTIN C.（编辑）技术研究和教育的最新进展。 跨学术界2017。智能系统和计算的进展，卷。660施普林格，湛，2018：289-296。https://doi.org/10.1007/978-3-319-67459-9_36
[8] CHEN J., LI K., TANG Z., BILAL K., YU S., WENG C., 和 LI K. 火花云计算环境下大数据的并行随机森林算法。IEEE并行和分布式系统汇刊, 2017, 28(4): 919-933。https://doi.org/10.1109/TPDS.2016.2603511
[9] SINGH S., & CHANA I. 地球：云计算中的能量感知自主资源调度。智能与模糊系统杂志, 2016, 30(3): 1581–1600. https://doi.org/10.3233/IFS-151866
[10] MA J., LI W., FU T., YAN L., 和 HU G. 基于改进遗传算法的云计算动态任务调度算法。在：ZENG Q. A. (编辑。) 无线通信、网络和应用。电气工程讲义，卷。348施普林格, 新德里, 2016年：829–835。https://doi.org/10.1007/978-81-322-2580-5_75
[11] WEI W., FAN X., SONG H., FAN X., 和 YANG J. 基于不完美信息动态斯塔克尔贝格博弈的基于隐马尔可夫的云计算资源分配。IEEE服务计算汇刊，2018年, 11(1)：78–89。https://doi.org/10.1109/TSC.2016.2528246
[12] PILLAI P. S., & RAO S. 使用博弈论不确定性原理的云计算资源分配。IEEE系统杂志，2016年, 10(2)：637–648。https://doi.org/10.1109/JSYST.2014.2314861
[13] SOTOMAYOR B.、MONTERO R. S.、LLORENTE I. M. 和 FOSTER I. 私有云和混合云中的虚拟基础架构管理。IEEE互联网计算，2009, 13（5）：14-22。https://doi.org/10.1109/MIC.2009.119
[14] RADOJEVIC B., & ZAGAR M. 托管（云）环境中负载平衡算法问题的分析。第34届国际会议米普罗会议记录，奥帕蒂亚，2011年，第 416-420页。https://ieeexplore.ieee.org/document/5967092
[15] LEE R., & JENG B. 云中的负载平衡策略。网络赋能分布式计算与知识发现国际会议论文集，北京，2011，第 447-454页。https://doi.org/10.1109/CyberC.2011.79
[16] WANG S. C., YAN K. Q., LAO W. P., 和 WANG S. S. 实现负载平衡三级云计算网络。第三届计算机科学与信息技术国际会议论文集，成都，2010，第 108-113页。https://doi.org/10.1109/ICCSIT.2010.5563889
[17] KADIM U. N. 和 MOHAMMED I. J. 云数据中心的基于混合软件定义网络的负载平衡和调度机制。西南交通大学学报, 2020, 55(3). https://doi.org/10.35741/issn.0258-2724.55.3.3
[18] ABDUL-JABBAR S. S.、ALDUJAILI A.、MOHAMMED S. G. 和 SAEED H. S. 云计算环境中的完整性和安全性：回顾。西南交通大学学报, 2020, 55(1). https://doi.org/10.35741/issn.0258-2724.55.1.11