Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 10, October 2021: 428-439

A survey on cloud-fog computing and load balancing scheduling algorithms

[1]Asha Sohal, [2]Ramesh Kait

Department of Computer science and applications, Research Scholar, Kurukshetra University, Kurukshetra
Department of Computer science and applications, Assistant Professor, Kurukshetra University, Kurukshetra

Corresponding Author[1]: asha.sohal24@gmail.com

Abstract

Cloud computing is a wide and fastest growing area in terms of computing research and industry these days. It mainly provides services based on IaaS, SaaS, and PaaS. These are the key parameters which decide the role of cloud services to the end users. These services can be offered to the end users through virtualization over the internet. Cloud has many advantages like large scaled computing, flexible infrastructures, pay per use, on demand services and many more. There are some major issues in processing of jobs over cloud computing like security, equal distribution of load, fault tolerance etc. and the biggest challenge over cloud is latency time which means the total time between the data sent by IoT over cloud, processing time and finally reply to the IoT or vice versa. The IoT devices will cause large number of various types of data that majorly could be difficult to handle for traditional system and sometimes over cloud systems to manage. Therefore, Fog computing is the paradigm which helps in upgradation of the current systems. It includes well equipped storage structure, computation, easily accessible data, and application-based services same as to the cloud computing closer to the users. In this survey paper, we introduce a comprehensive summary on the fog cloud networks, its architecture and the comparative study of load balancing algorithms which make fog cloud systems more efficient, and the intensive role of fog devices are like to act in advanced areas as in Tactile Internet.

Keywords: Cloud computing, fog computing, End Users, load balancing

Introduction

Over the years, computing paradigms have evolved from distributed, parallel, and grid to cloud computing. These days cloud computing plays important role in online data storage so we can say that cloud computing have many inherent features like elasticity in nature, on-request service, or allocation of resources, reduced efforts of management, flexible in pricing model based on pay-as-per-use, easy to use applications and provisioning of servers. Basically, its categories into three types of cloud services models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) and anything as a service (XaaS). All the services based on the concept of virtualization. IaaS includes the features regarding virtualized resources, computation, storage capacity, and networking (Stojmenovic, 2015). The PaaS presents software-based environments or platforms for the enhancement, utilization, and supervision of applications. This delivers applications based on software and SaaS combines services to end users' and other relevant applications. The XaaS provides the information apart from IaaS, PaaS, and SaaS. Nowadays, there are many cloud service providers with several advantages. However, the system has some major pitfalls and crucial drawback is the connection in the middle of cloud and the end suers' machines (Bonomi et al., 2012). Although large scale network connectivity is set all over the Internet which is sometimes not appropriate for a huge set of cloud-based applications like latency-sensitive based data etc. There are many well-known examples in which cloud setup fails which includes vehicles in connected form, detection of fire and firefighting, smart grid structure, and delivery of content. Moreover, cloud-based products are frequently following disseminated architecture and also consists of multiple components. Therefore, the situation is not ordinary sometimes to deploy application modules separately over multiple cloud setup. This may deteriorate the latency time due to the operating cost produced by inter-cloud communications between them. Another restriction is the directives may prescribe administering at locations where the cloud service provider may have no data centers. To work on these limitations, we found a solution that is by considering fog layer in between the cloud system and end users/IoTs. It gives a better explanation to cloud platform with better efficiency (Guevara & da Fonseca, 2021). In fig 1, we showed three-layer architecture of fog-cloud networks which includes cloud layer, edge data centres, edge devices and fog nodes layer. We considered data between cores to edge. Generally, we can say that Fog is "cloud closer to ground" and we can easily share and fetch our data over cloud time to time without any hurdles (Sohal & Kait, 2020). There are many ways to do the same in efficient way. With the help of fog computing, things can be accessed easily very closer to the network like latency based sensitive data and some data like delay-tolerant and computational insensitive data can be processed over the cloud. However, we can say that fog layer helps in reducing the load and power consumption over the cloud. The most important things over cloud are computing, data storage, and networking services and the fog extends all the facilities near to edge which enhances the performance of all IoTs (Sohal & Kait, 2021). Moreover, the fog-based layer provides extra valuable advantages, such as less latency rate, execution and availability of things are closer to the edge of users which is known as fog nodes. They also help in processing of data at specific locations rather than on data centres. It follows the concept of virtualization and distributed system sothat there should be maintain equilibrium at the end of users.

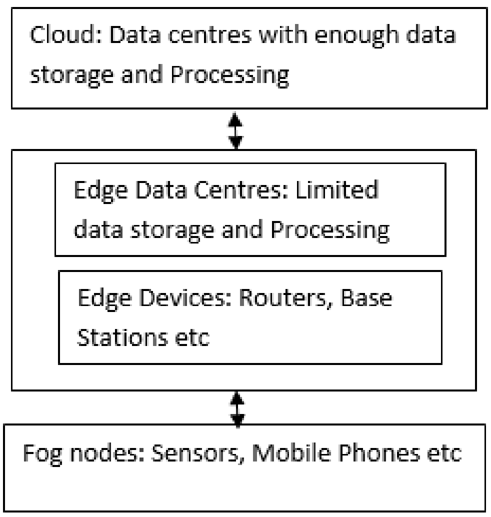

Figure 1. Three Layer Architecture of Fog-Cloud Networks

Majorly process, storage system, and networking essential services are the developing zones of the cloud setup and the fog system that broadens it services (Vaquero & Rodero-Merino, 2014). Still, the fog system delivers some supplementary benefits such as less latency time, by permitting execution near to the network edge, near to the end users' devices, jobs handle by the fog nodes and the capability to enable execution at specific locations. The system also extends densely and equally allocated points for collecting the data which is generated by the end users' devices. This may happen because of proxies servers, nearby access points, and routers located at the network edge, near the sources as shown in fig 2 (Yi et al., 2015). This shows how the communication takes place over cloud and fog.

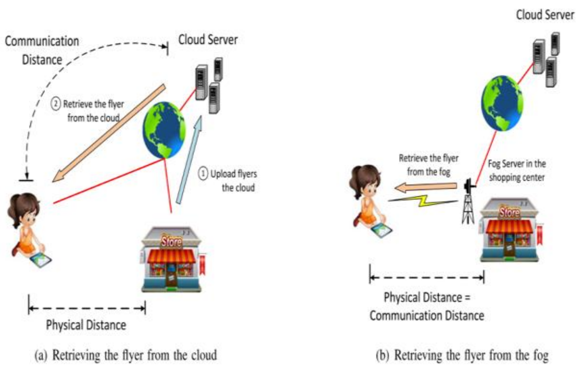

Figure 2. Retrieval of data from cloud and fog

Numerous surveys and discussion group related to the fog computing had been circulated over the past years. A classification scheme of load balancing algorithms in cloud system is shown in figure 3. This review paper boards several groups of readers: readers those who are interested in detailed history of fog, readers have keen interest in architectural viewpoints, and readers interested in general applications.

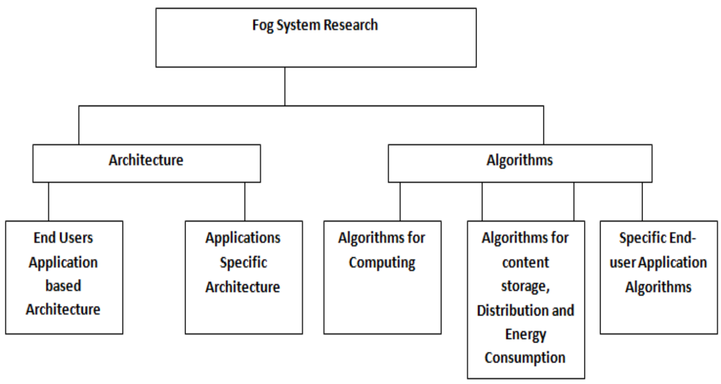

Figure 3. Classification of Fog System Research

Figure 3 illustrated the brief detail about fog system research, its architecture, and related algorithms. The detailed literature survey on the fog computing is highly distinct and structuring similar works in a methodical way is not an insignificant task (Stojmenovic & Wen, n.d.,2014). Generally, fog computing can be characterized into projected structural design for the fog systems and intended algorithms for the fog systems. The individual assessment of these two perspectives was a natural preference, because most of the researchers in this field tackles the concern from any of the point of view. It mainly allows grouping the evaluated papers under common umbrellas. In first part of the system of structural design for fog systems, there are two subdivisions which are architectures based on application agnostic and architectures based on application specific. Other part of the system of algorithms in fog systems, there are four subsections which are computing based algorithms, content storage-based algorithms and dissemination and energy consumption, and algorithms based on application-specific. Based on algorithms and architecture, we can provide an interface in between Fog and Cloud Computing. Moreover, Fog computing nomenclature was borne from the fact that fog is a cloud close to the ground, intending to bring cloud computing much closer to Internet of Things. The advent of IoT has caused in several use cases which generates a substantial volume of data, combining the challenges of dealing with big data from many geographically distributed data sources (Stojmenovic et al., 2016). To efficiently scrutinize these time-sensitive statistics, the upcoming technique 'fog computing' was proposed by CISCO. To harness the gains of IoT and speed up the knowledge and response to events, we expect a new set of infrastructures as the existing cloud models are not intended to handle the specifics of IoT (i.e., volume, variety, and velocity of data). Specifically, billions of previously unconnected devices are now generating over two Exabyte's of data every day, and it has been estimated that by 2020, 50 billion "things" will be connected to the Internet. Therefore, fog computing has been identified as a viable solution.

Literature survey

Guevara, Judy et al. discussed about concept of two schedulers based on integer linear programming which is used to handle different tasks schedule either in the resources of cloud or on fog (Guevara & da Fonseca, 2021). In this paper, they have used random and round robin algorithms to get numerical results which perform efficiently without causing the violation of QoS requirements.

Srirama, Satish et al. described about the proposed frame for the Akka disseminated Fog applications based on the Actor Model. They also mentioned about the detailed survey on network of wireless sensors which was used to find the feasibility of designing applications on the Fog networks to test the proposed actor-based structure (Srirama et al., 2021).

Abedi and Pourkiani illustrated the techniques of minimization of internet load and their response time by disbursing the jobs in between cloud and fog servers. They also discussed about some modified algorithms which helps in significant reduction in the internet load and response time in parallel to some fog and cloud-driven methods (Abedi & Pourkiani, 2020). However, the result of this proposed algorithm is pronounced when the number of jobs required in the broker is increased that means all the jobs directly sent to the servers for execution. The simulation has been done using MATLAB.

Jindal et al. they proposed an approach which could take a better decision on time about when the jobs should be transferred from fog servers to cloud servers. This happened when there was lack of handling all jobs over fog nodes individually because of less capacity to process, complex calculations etc. After that, tasks would be sent to cloud servers for further execution using FogSim (Jindal et al., n.d., 2020). The bandwidth used to unload the Fog data is generally less than the bandwidth needed for the server to discharge the data. Besides, leasing cloud data center services and their capacities are higher than Fog node prices and capacities.

Delfin et al. discussed about how to analyze the focal points used in administrations of fog computing and also observed that Fog computing is similar to the cloud, where as the main gap lies in the technique that has been found much closer to the end users' devices to access and provide a reaction to the client in a very less stretch of time (Delfin et al., 2019). They used CISCO tool to examine the results in a better way.

Rabay'a et al. defined the use of simulation tool named as PeerfactSim.KOM. To quantify the theoretical Fog P2P model, which demonstrated that the cloud and Fog computing architectures' bandwidth efficiency was superior with the file transfer method(*Fog Computing with P2P: Enhancing Fog Computing Bandwidth for IoT Scenarios | IEEE Conference Publication | IEEE Xplore*, n.d., 2019). The findings of the study offered a clear base for potential attempts to refine p2p Fog computing.

Ali et al. described about the comparison and technicality of fog computing and cloud computing. They explored more flexible and higher service in data processing rate in fog computing than cloud computing. This also consumed very low network bandwidth instead of transferring all data over the cloud. The analysis has been done using CloudSim in this paper (Kumar et al., 2019).

Ema et al. discussed about how to provide Fog infrastructure facilities in real-time—concentrating on addressing cloud store problems using blockchain technology. They explored that the conventional cloud data center cannot support a massive volume of data storage because it is too pricey and requires time. They also found that Fog computation can manage load and reduce the conventional cloud data center burden(Ema et al., 2019).

LM Vaquero et al. illustrated the detailed survey on the emerging topics of fog computing in terms of its real life applications and emerging prospects in usage patterns(Vaquero & Rodero-Merino, 2014). There are certain challenges discussed in the paper which can be further explored for better results.

Yi et al. defined about the main functioning of fog computing and its related concepts such as cloud computing (Yi et al., 2015). They introduced application-based circumstances and discussed the challenges as well and leave a scope for future aspects.

Chiang et al. demonstrated a historical study on fog computing. In this paper, they compared at a very high point about the differences between fog, edge, and cloud computing. They additionally presented the improvements of fog computing and leave a scope for research challenges [(Chiang & Zhang, 2016)]. Many studies have also been distributed on fog computing at enormous scale and in the context of particular application spheres, i.e., vehicular Ad-hoc Networks (VANETs), Radio Access Networks (RAN) and Internet of Things (IoTs).

Fang et al. discussed about how to set balanced load using two-level scheduling approach. They also used scheduling approach which selected the random customer requirements and tried to balance the equal load over the cloud surface (Fang et al., 2010). This satisfies the users requirements in random way and showed increase in resource utilization. Furthermore, we can increase in count of users' requirements in terms of data bandwidth, data cost etc. in future prospect.

Kun Li et al. described how to schedule each task over cloud to get balanced load using LBACO (load Balancing Ant Colony Optimization) technique. The most important key point of the document is to proportionate the total work capacity of the scheme and to decrease the makespan of a assigned set of tasks (Li et al., 2011). They simulated their results on CloudSim and found that load Balancing Ant Colony Optimization technique performed well than FCFS (First Come First Serve) algorithm and ACO (Ant Colony Optimization) algorithm. In future, we can do simulation on the heterogeneous tasks and try to find more details of users' requirements.

Jinhua Hu et al. proposed a technique to balance the load of virtual machine resources using genetic algorithm. As per the existing data and the present scenario of the system and with the help of genetic algorithm, they were able to deploy the required virtual machine resources and chosen the least-affective way out by which they targeted the safest load balancing results and minimized the dynamic movement. They also tried to solve the unequal load distribution and minimize the migration cost using this strategy. These types of techniques helped in maintain when system is stable and dynamic (Hu et al., 2010). They concluded to mention the various changes in virtual machines load and processing cost that can be managed in future work by proper analyzing and monitoring mechanism.

Pandey et al. discussed about a heuristic approach to minimize the computation cost and data transfer cost using particle swarm optimization (PSO). They scheduled services and applications in such a way sothat results showed maximum productivity of the work. For better result analysis, they have considered data in dynamic mode. After that they compared their results with BRS (Best Resource Selection) algorithm and proved their own results are three times better than Best Resource Selection and distribution of load was quite satisfactory among all resources (Pandey et al., 2010). This work will increase its performance by using integrated particle swarm optimization based heuristic approach into the existing system and workflows based on real life applications setup like in analysis in brain imagination.

Ge et al. premeditated a modified scheduler technique which accepted a scheduling pronouncement by observing the complete cluster of jobs in the job succession (Ge & Wei, 2010). In this paper, they have used optimizing techniques in scheduler and suggested the use of genetic algorithm to get better results. While using this scheduler, there was less chance to process all the jobs simultaneously and there were certain hurdles in finding the proper time of execution and delay in predicting of computation time.

Xiaonian Wu et al. proposed an technique in cloud computing based on the quality of service-driven approach which was known as task scheduling technique (Wu et al., 2013). In this technique, they included quality of service-driven applications like user opportunities, expectancy, length of the tasks and the waiting time in the queue. All jobs have been scheduled in the queue according to their increasing order of job completion time. The results indicated a clean graph how the load assigned to each server in a distributed way, and it included quality of service by considering urgency and the accomplishment time of jobs as well.

Sindhu et al. defined two different algorithms which are efficient task scheduling in fog-cloud system named as longest cloudlet fastest processing elements (LCFP) and shortest cloudlet fastest processing elements (SCFP). They designed an interface which helped in reduction of time used in case of turnaround, increase in utilization of resources and complexity in computational (Sindhu & Mukherjee, 2011). The future scope of this paper is to enhance quality of service of fog-cloud grids by using augmented techniques.

Raju et al. discussed about hybrid algorithm to minimize the job completion time (Raju et al., 2013). This algorithm included the major positive results of the ACO (Ant Colony Optimization) and search based on Cuckoo. Also, this algorithm was compared with ant colony optimization about its performance in terms of load balancing and energy consumption. We can further include some other similar algorithms to increase its work over utilization of several tasks at a time.

Chen et al. considered a set of tasks which was present in a queue to schedule them using an algorithm named as "Min-Min Scheduling". In this algorithm, a source point has been selected based on minimum time for processing among all the jobs. It also considered the job which was in smaller in size and that job has been scheduled for final processing. This was the phenomenon in Min-Min scheduling algorithm (Chen et al., 2013). After the completion of first task, it was emptied from the group of tasks and the same algorithm was reprocessed for the other tasks till the end which were present in the set. The future scope of this paper to increase in load balancing techniques because existing algorithm failed to find the source in some cases.

Jia Zhao et al. mentioned about balanced load techniques by considering algorithms of different fields which tried to optimize the candidate targeted host after that it was also employed to predict instantaneous effect in load balancing by choosing up the optimum targeted host (Zhao et al., 2016). In this immediate effect of balanced load, there was less chance to get high execution efficiency for other tasks because of high utilization of relevant resources using balanced load based on Bayes and Clustering (LB-BC). It consisted of long-term activity for achieving global ratio of balanced load in comparison to the propertied used in effect of immediate load balancing. Furthermore, we can design the layout of load balancing techniques using wide area networks.

Chun-wei et al. aimed a hyper heuristic scheduling algorithms to enhance the implementation of rule-based scheduling algorithms. The proposed work combines various heuristic algorithms including genetic algorithm, ant-colony optimizations, simulated annealing, and particle swarm optimization. discussed about cloud computing system to find out best scheduling solutions. To fine tune the scheduling algorithms two operators are recommended (Tsai et al., 2014). In comparison to other scheduling algorithms, HHSA tried to reduce execution time and task scheduling processing time.

Liyun et al. proposed a multi-objective optimization and resource cost model for handling task-based scheduling problems. They considered the detailed request of jobs based on resources available. These types of models provided the relationship in between the resource costs of multiple users and total budget costs. To analyze this problem in proper way, they also used modified ant colony algorithm. In this paper, demand of requests for resources was defined by user resources cost (Zuo et al., n.d.,2015). Also discussed about how the multi-objective optimization approach is better than first come first serve and min-min algorithm at incurred value.

Comparison in scheduling techniques of load balancing

There is an increase in online data storage and a burden of data processing, storing, and transferring over fog-cloud network is difficult these days**.** To maintain the healthy system over fog-cloud system, we must share the load equally on numerous servers for rapid execution of work and provides cost-effective resource utilization. By considering these types of scheduling techniques, energy can be absorbed during processing of jobs in a better way. There are countless types of load balancing scheduling techniques:

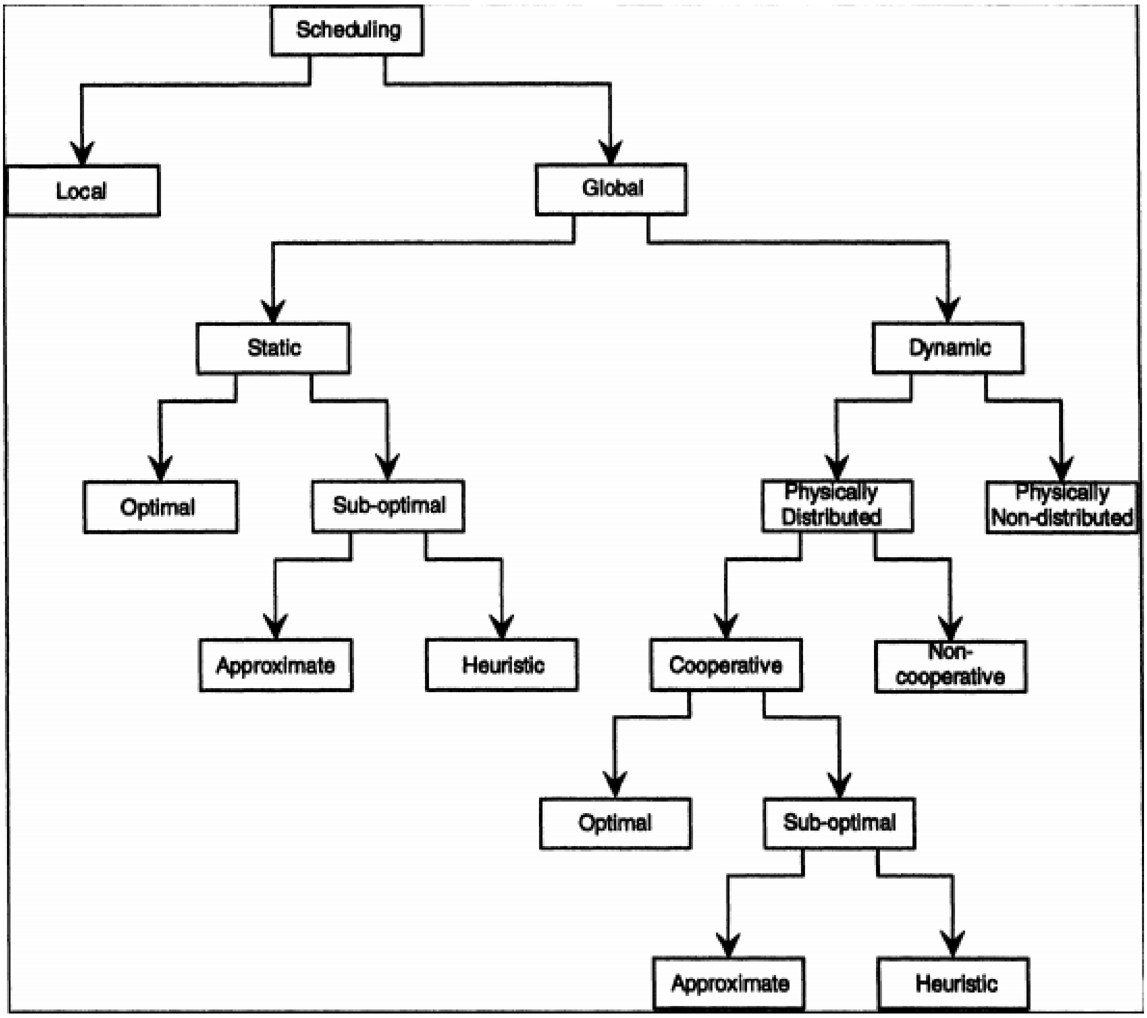

Figure 4. Classification of Scheduling Techniques in Load Balancing

In case of static scheduling, the job allocation of tasks for processing is accomplished before the implementation of program begins. The material related to performance of task and its administering resources is supposed to be consider at the moment of compilation. The assigned task is executed on the allotted processes that is static scheduling practices are processors non-preemptive in nature. Basically, the key motive of static scheduling practices is to reduce the total processing time of a simultaneous program while reducing the communication interruptions. These techniques further categorized into optimal technique and suboptimal technique. Regrettably, one of the most vital inadequacies of static scheduling technique is that, to generate optimum schedules is an NP-complete challenge because this generates optimal results in limited cases. Furthermore, these approaches are classed into approximate and heuristic approaches. In case of approximate suboptimal static scheduling techniques, there is a searching method for finding the solution space either a depth-first fashion or a breadth-first. Nevertheless, instead of exploring the entire solution space for an optimum solution, this technique halts when an appropriate solution is attained. As the name defines "Heuristic methods" it depends on rules-of-thumb to steer the scheduling activity in the right way to reach a "near" optimum result. In dynamic scheduling, it depends on the restructuring of processes in execution time amongst the processors. This sort of restructuring is measured by assigning each task with the help of severely overloaded processors to the lightly loaded processors which is known as load balancing. Another approach is used to improve the performance of the application.

The significance of dynamic scheduling on static scheduling is that there is lack of awareness of system behavior at the run time of the requests before execution. The tractability intrinsic in dynamic load balancing permits for adaptation to the unpredicted application requirements at run-time. Also, dynamic scheduling is primarily useful in a system which comprises of a network of workstations in which the most important performance goal line is to maximize consumption of the processing energy instead of minimizing execution time of the applications. The foremost limitations of dynamic scheduling techniques are the run-time operating cost due to:

• the handover of load information among processors

• the collection process of decision-making processes and handing over the job among processors

• the delay in transmission because of task relocation itself.

Furthermore, Table 1 illustrated the summarized details of publications of research papers.

Table 1

Summary of Research Papers

Title	**Year**	**Title**
Task scheduling in cloud-fog computing systems	2021	Qubahan Academic Journal
Akka framework based on the Actor model for executing distributed Fog Computing applications	2021	Future Generation Computer Systems
Resource Allocation in Combined Fog-Cloud Scenarios by Using Artificial Intelligence	2020	International Conference
MTFCT: A task offloading approach for fog computing and cloud computing	2020	Qubahan Academic Journal
Fog computing: A new era of cloud computing	2019	Scopus
A Study of Moving from Cloud Computing to Fog Computing	2019	Qubahan Academic Journal
Suitability of Using Fog Computing Alongside Cloud Computing	2019	Scopus
A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing	2010	Springer
Cloud task scheduling based on Load Balancing Ant Colony optimization	2011	IEEE

A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment	2011	IEEE
Honeybee behavior inspired load balancing of tasks in cloud computing environments	2013	ACM
GA-Based Task Scheduler for the Cloud Computing Systems	2010	IEEE
A task scheduling algorithm based on QoS driven in cloud Computing	2013	ACM
Efficient task scheduling algorithms for cloud computing environments	2011	Springer
Minimize the make span using Hybrid algorithm for cloud computing	2013	IEEE
User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing	2013	IEEE
A Heuristic Clustering Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment	2016	IEEE Transaction
A hyper-heuristic scheduling algorithm for cloud	2014	IEEE Transaction
A Multi-Objective Optimization Scheduling Method Based on the ACO in Cloud Computing	2015	IEEE Access

Conclusion

In this survey paper, we deliberated about how the emerging fog computing has the capacity to handle the users' request and process subsequently the unprecedented amount of data from end users or IoTs. Whereas cloud computing can handle all the on-demand request from users' side and fulfil their requirement by providing related services, however applications or services that need low latency time and high quality of service (QoS) will not be able to perform smoothly because of congestion over the cloud. In this situation, Fog computing supports cloud computing to make available the fog nodes near to the edge of the users sothat there are less chances of congestion and for better productivity. With the help of several load balancing algorithms and scheduling techniques, we can assure the users for ease in the access of fog-cloud services.

The characteristics of fog computing minimizes the congestion of data on the cloud, improves latency time and its performance. Furthermore, there are certain major issues that need to be address in future perspective, such as more secured network, privacy, better resource utilization, monetization, and ease of provisioning servers.

References

1. Abedi, M., & Pourkiani, M. (2020). Resource Allocation in Combined Fog-Cloud Scenarios by Using Artificial

Intelligence. 2020 5th International Conference on Fog and Mobile Edge Computing, FMEC 2020, 218–222. https://doi.org/10.1109/FMEC49853.2020.9144693
2. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog Computing and Its Role in the Internet of Things. Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing - MCC '12. https://doi.org/10.1145/2342509
3. Chen, H., Wang, F., Helian, N., & Akanmu, G. (2013). User-priority guided min-min scheduling algorithm for load balancing in cloud computing. 2013 National Conference on Parallel Computing Technologies, PARCOMPTECH 2013. https://doi.org/10.1109/PARCOMPTECH.2013.6621389
4. Chiang, M., & Zhang, T. (2016). Fog and IoT: An Overview of Research Opportunities. IEEE Internet of Things Journal, 3(6), 854–864. https://doi.org/10.1109/JIOT.2016.2584538
5. Delfin, S., Sivasanker, N. P., Anand, A., & Raj, N. (2019). Fog computing: A new era of cloud computing. Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019, 1106–1111. https://doi.org/10.1109/ICCMC.2019.8819633
6. Ema, R. R., Islam, T., & Ahmed, M. H. (2019, July 1). Suitability of Using Fog Computing Alongside Cloud Computing. 2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019. https://doi.org/10.1109/ICCCNT45670.2019.8944906
7. Fang, Y., Wang, F., & Ge, J. (2010). A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6318 LNCS(M4D), 271–277. https://doi.org/10.1007/978-3-642-16515-3_34
8. Fog Computing with P2P: Enhancing Fog Computing Bandwidth for IoT Scenarios | IEEE Conference Publication | IEEE Xplore. (n.d.). Retrieved July 15, 2021, from https://ieeexplore.ieee.org/document/8875437
9. Ge, Y., & Wei, G. (2010). GA-based task scheduler for the cloud computing systems. Proceedings - 2010 International Conference on Web Information Systems and Mining, WISM 2010, 2, 181–186. https://doi.org/10.1109/WISM.2010.87
10. Guevara, J. C., & da Fonseca, N. L. S. (2021). Task scheduling in cloud-fog computing systems. Peer-to-Peer Networking and Applications, 14(2), 962–977. https://doi.org/10.1007/S12083-020-01051-9
11. Hu, J., Gu, J., Sun, G., & Zhao, T. (2010). A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. Proceedings - 3rd International Symposium on Parallel Architectures, Algorithms and Programming, PAAP 2010, 89–96. https://doi.org/10.1109/PAAP.2010.65
12. Jindal, R., Kumar, N., & Nirwan, H. (n.d.). MTFCT: A task offloading approach for fog computing and cloud computing.
13. Kumar, V., Laghari, A. A., Karim, S., Shakir, M., & Anwar Brohi, A. (2019). Comparison of Fog Computing & Cloud Computing. International Journal of Mathematical Sciences and Computing, 5(1), 31–41. https://doi.org/10.5815/IJMSC.2019.01.03
14. Li, K., Xu, G., Zhao, G., Dong, Y., & Wang, D. (2011). Cloud Task scheduling based on Load Balancing Ant Colony Optimization. https://doi.org/10.1109/ChinaGrid.2011.17
15. Pandey, S., Wu, L., Guru, S. M., & Buyya, R. (2010). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. Proceedings - International Conference on Advanced Information Networking and Applications, AINA, 400–407. https://doi.org/10.1109/AINA.2010.31
16. Raju, R., Babukarthik, R. G., Chandramohan, D., Dhavachelvan, P., & Vengattaraman, T. (2013). Minimizing the makespan using Hybrid algorithm for cloud computing. Proceedings of the 2013 3rd IEEE International Advance Computing Conference, IACC 2013, 957–962. https://doi.org/10.1109/IADCC.2013.6514356
17. Sindhu, S., & Mukherjee, S. (2011). Efficient Task Scheduling Algorithms for Cloud Computing Environment. Communications in Computer and Information Science, 169 CCIS, 79–83. https://doi.org/10.1007/978-3-642-22577-2_11
18. Sohal, A., & Kait, R. (2020). Review On Optimal Mathematical Workload Allocation Models In Energy Consumption Using Fog-Cloud Networks. SSRN Electronic Journal. https://doi.org/10.2139/SSRN.3565858
19. Sohal, A., & Kait, R. (2021). Sohal, A., & Kait, R. (2021). COMPARATIVE STUDY OF OPTIMIZED LOAD BALANCING MODELS USING FOG-CLOUD NETWORKS. Gurugram International Journal of Technical Research. https://doi.org/10.30780/specialissue-ICAASET021/003c. Gurugram International Journal of Technical Research. https://doi.org/10.30780/specialissue-ICAASET021/003
20. Srirama, S. N., Dick, F. M. S., & Adhikari, M. (2021). Akka framework based on the Actor model for executing

distributed Fog Computing applications. Future Generation Computer Systems, 117, 439–452. https://doi.org/10.1016/J.FUTURE.2020.12.011

21. Stojmenovic, I. (2015). Fog computing: A cloud to the ground support for smart things and machine-to-machine networks. 2014 Australasian Telecommunication Networks and Applications Conference, ATNAC 2014, 117–122. https://doi.org/10.1109/ATNAC.2014.7020884
22. Stojmenovic, I., & Wen, S. (n.d.). The Fog Computing Paradigm: Scenarios and Security Issues. https://doi.org/10.15439/2014F503
23. Stojmenovic, I., Wen, S., Huang, X., & Luan, H. (2016). An overview of Fog computing and its security issues. Concurrency and Computation: Practice and Experience, 28(10), 2991–3005. https://doi.org/10.1002/CPE.3485
24. Tsai, C.-W., Huang, W.-C., Chiang, M.-H., Chiang, M.-C., & Yang, C.-S. (2014). A Hyper-Heuristic Scheduling Algorithm for Cloud. IEEE Transactions on Cloud Computing, 2(2), 236–250. https://doi.org/10.1109/TCC.2014.2315797
25. Vaquero, L. M., & Rodero-Merino, L. (2014). Finding your way in the fog: Towards a comprehensive definition of fog computing. Computer Communication Review, 44(5), 27–32. https://doi.org/10.1145/2677046.2677052
26. Wu, X., Deng, M., Zhang, R., Zeng, B., & Zhou, S. (2013). A Task Scheduling Algorithm based on QoS-Driven in Cloud Computing. Procedia Computer Science, 17, 1162–1169. https://doi.org/10.1016/J.PROCS.2013.05.148
27. Yi, S., Li, C., & Li, Q. (2015). A survey of fog computing: Concepts, applications and issues. Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2015-June, 37–42. https://doi.org/10.1145/2757384.2757397
28. Zhao, J., Yang, K., Wei, X., Ding, Y., Hu, L., & Xu, G. (2016). A Heuristic Clustering-Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment. IEEE Transactions on Parallel and Distributed Systems, 27(2), 305–316. https://doi.org/10.1109/TPDS.2015.2402655
29. Zuo, L., Shu, L., Dong, S., Zhu, C., & Hara, T. (n.d.). SPECIAL SECTION ON BIG DATA SERVICES AND COMPUTATIONAL INTELLIGENCE FOR INDUSTRIAL SYSTEMS A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing. https://doi.org/10.1109/ACCESS.2015.2508940