

Survey on Load Balancing and Scheduling Algorithms in Cloud Integrated Fog Environment

Jyoti Bisht¹ and Dr. V V Subrahmanyam²

¹Research Scholar, School of Computer and Information Sciences, IGNOU, New Delhi, INDIA
{Email: Jyotibisht8891@gmail.com}

²Director, School of Computer and Information Sciences, IGNOU, New Delhi, INDIA
{Email: vvsbrahmanyam@ignou.ac.in}

Abstract- As technologies like IoT are emerging, increasing number of users on internet and workload on Cloud is increasing at an enormous rate along with the requirements of network bandwidth. Also the increasing demand for real time analytics and real time applications lead to the requirement of a new technology. Fog Computing in collaboration with cloud computing can provide a solution for all such requirements. This technology brings cloud computing resources like memory, computation near end users and hence provides a solution. End devices like gateways, routers, cell base stations can be used as fog devices. Since these devices have limited resources, soon they will get overloaded. Hence proper scheduling and load balancing mechanism is needed to fully utilize the benefits of this technology. Various algorithms proposed to resolve the issue of task scheduling, resource provisioning and load balancing are discussed in this paper. We have also identified some of their shortcomings for further development.

Keywords- Fog computing, Cloud Computing, Scheduling, Load balancing.

1 Introduction

In the age of information technology, data is major commodity, and way of possessing and managing this data tends to generate more data driven businesses value. Cisco estimated that 50 billion devices will be connected by 2020 [1] with the increasing use of Internet of Things (IoT) and hence will lead to massive amount of data generation. It also estimated in its white paper that data center IP traffic will reach 20.6 Zettabytes per year by the end of 2021 from 6.8 ZB per year in 2016 as is given in Fig 1. It is also estimated that social networking (26% CAGR from 2016 to 2021) and video streaming (24 percent CAGR from 2016 to 2021) will be the fastest growing applications over internet. Also, the total amount of data created by any device will reach 847 ZB per year by 2021 from 218 ZB per year in 2016 due to emerging popularity of Internet of Things [2]. Another paper by Cisco Visual Networking Index (VNI) estimated overall IP traffic to grow from 122 EB per month in 2017 to 396 EB per month by 2022 [3] as given in Fig 2.

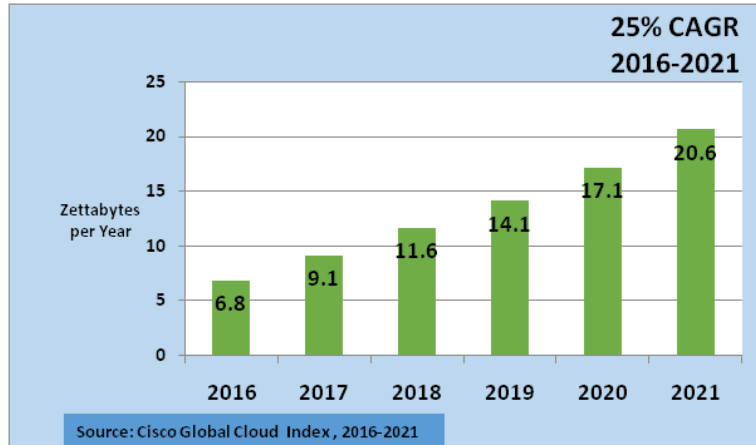


Fig. 1. Global data center IP traffic growth

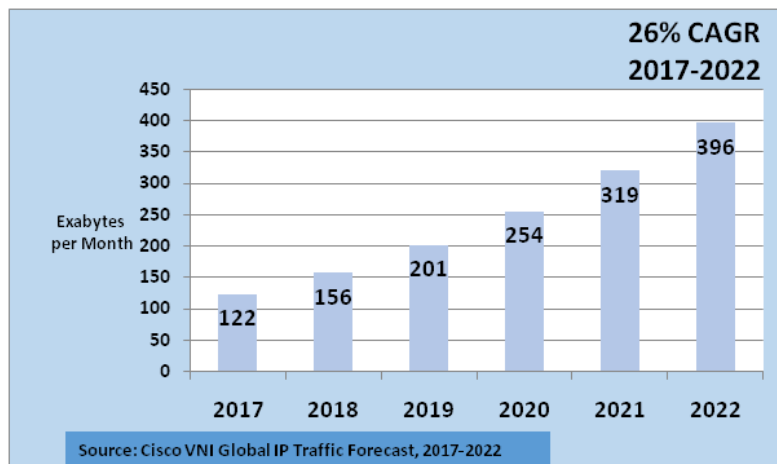


Fig. 2. Overall IP traffic expected to grow

Cloud offers a solution by delivering reliable services to its users by the use of data centers based on virtualization technology in which a single machine can act as multiple virtual machines and hence can handle requests from multiple users at a time independently. Consumers from anywhere in the world can access services provided by cloud on demand. Instead of consumers owning resources and managing themselves, current pay per use Cloud computing technology is an efficient solution since customers don't have to pay for initial setup cost and also not have to bother about the management, networking, licensing, technical issues and others. However, this technology is not suitable for latency-sensitive applications like monitoring of patients, real-time applications, traffic management, self-driving cars, cognitive assistance, since it requires computing nodes to be present nearby to fulfil the request within some time constraints. It is difficult for cloud computing technology to satisfy

requirements of low latency, mobility, location awareness and other applications that demands real time responses, since cloud servers are centralized and remotely located.

It would be infeasible and inefficient to use cloud in many applications due to bandwidth constraints since volume and velocity of data, which are two identified characteristics of Bigdata is increasing day by day. Also it would not be appropriate solution for many applications like email, photo storing, online medical records, and others due to privacy concerns.

1.1. Fog Computing and its Architecture

In January 2014, Cisco given a term Fog Computing to a technology that extends computing to the edge of the network. It is a way to bring cloud computing capabilities to the edge of the network. Fog computing can act as a middle layer between the cloud and IoT devices. It can provideservices of computing, storage, networking, and data management within the close vicinity of IoT devices and users on the network. It can be called as an extension of cloud computing. Benefits provided by cloud can still be provided with these extensions to fog, which includes, manageability, containerization, virtualization, orchestration, and others but with better efficiency.

OpenFog Consortium or at present known as Industrial Internet Consortiumdefined fog computing as “a horizontal system level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum” [4].

Traditional cloud computing model can continue to remain as a centralized computing systems having major responsibilities with few capabilities shifted towards or offered by fog devices as a distributed system. The selection or decision of which tasks should go to fog layer and which tasks should be forwarded to the backend cloud are specific to applications and depends upon the resource requirements and availability of resources and time constraints. These decisions could be taken statically or can change dynamically depending upon network state changes in areas like load onavailable storage, processor, bandwidth, fault events, security, and cost.

A three-layer generalized logical architecture for fog computing is depicted in Fig 3. The three layers are IoT or end devices at end layer, Fog servers at fog layer as middle layer, and Cloud servers at the cloud layer.

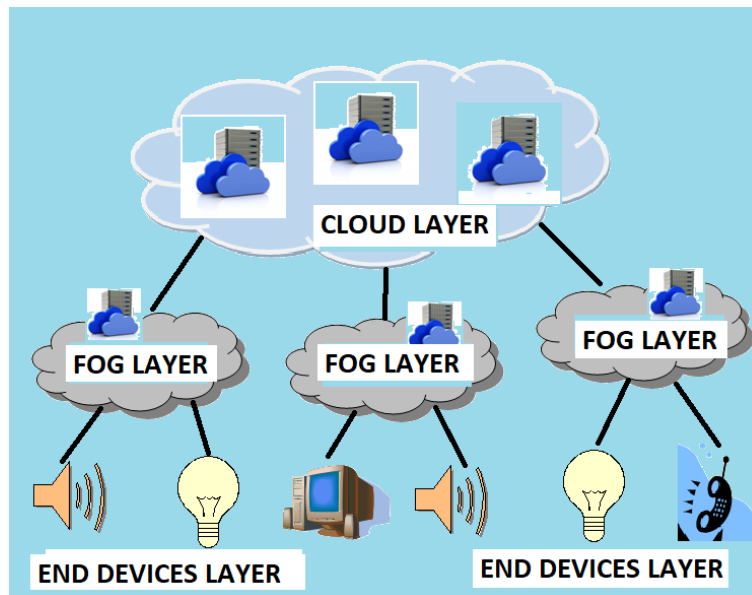


Fig 3: Fog Computing Architecture

A non proprietary fog computing architecture called Open Fog architecture is also established to support existing cloud computing services in addition to diverse IoT and edge-oriented ecosystem and identified scalability, agility, security, openness, among other “pillars” of open fog architecture [4]. IEEE Standards Association later adopted Open Fog Consortium’s reference architecture as a standard for fog computing through IEEE 1934[6].

The various characteristic of fog computing identified are edge location, location awareness, geographical distribution, low latency, large-scale sensor networks, mobility support, real-time computations, heterogeneity, wireless access, interoperability and federation[7] and its application scenarios are Smart Grid, Smart Traffic Lights, Wireless Sensor and Actuator Networks and Connected Vehicles, Decentralized Smart Building Control [5,8-9].

However, several obstacles are there which are yet unsolved research challenges and are needed to be solved to get full benefit from this emerging technology. Research challenges identified are - general purpose computing on fog nodes, discovering edge nodes, task placement, application placement, partitioning of tasks and offloading, improving quality-of-service (QoS) and experience (QoE), security and privacy issues[8,10].

2 Load Balancing and Scheduling in Fog computing

As discussed above, Fog computing architecture has three layers- end devices, middleware fog layer and centralized cloud servers at cloud layer. A fog layer in fog computing architecture

consists of several fog servers, where each fog server can contains multiple virtual machines. Also, each fog server like a cloud server must have processing power, communication unit and data storing capabilities.

Users can communicate directly with fog servers using wireless interface like Bluetooth, WI-Fi etc since they are present in the vicinity of users. Therefore, fog servers independent of the cloud servers, should provide pre-defined applications and pre-cached information to benefit users. Also, fog servers are connected to cloud servers by using wired or wireless connections so that users can have access to more services and resources available at cloud since limited resources are available at Fog servers.

Fog servers like cloud servers are based on virtualization technology. Multiple users request can be served by multiple virtual machines created in fog servers. If larger request arrives at fog layer, the request can be decomposed into a set of tasks, which can be then assigned to a several virtual machines. Decision of allocation of task to which virtual machine is taken by a scheduling technique decided by fog server. The aim of job or task scheduling technique is to optimally assign various tasks to virtual machines at fog layer with minimum use of resources like shortest execution time, least memory use. The job scheduling problem is NP-hard problem since there is no deterministic polynomial algorithm to solve it.

Also, multiple tasks can be submitted to a fog node from multiple users. It is also possible that large number of requests can arrive at a particular server forcing it to become overloaded soon in a situation like peak hours, while other fog servers are idle or lightly loaded. Hence proper balancing of load between servers and placement decision of servers are important for successful implementation of Fog Computing.

2.1 Categorization of Load Distributing Algorithms

Load distribution can improve the performance of fog computing by transferring tasks from fog server which are overloaded or service quality is reducing due to over burdening, to servers which are available to serve more requests or can provide better quality of service (QoS) due to availability of more or free resources. By this tasks can utilize resources that would otherwise go waste. This can lead to better performance and efficient utilization of resources. Load-distribution algorithms can be implemented in three ways- dynamic, static and adaptive.

Static load distribution algorithms do not make use of system's state information i.e. load on system. Based on previous knowledge about the system, decisions are taken. These algorithms can hence lead to poor decisions making because they do not consider node's state when making such decisions. It is also possible that a task which initially arrived at an idle node is transferred to a node which is already heavily loaded.

Dynamic load distribution algorithms use system's state information to make decisions. These algorithms must collect previous information, store it, analyze it and also make use of current load on the system. Use of state information can improve quality of decisions hence dynamic algorithms have the ability to perform better than static algorithms. But maintaining state information and analysing that data can cause overhead to the system.

Adaptive load-distribution algorithms are a special type of dynamic algorithms. According to the changing system state, they can dynamically adapt new policies and change their parameters i.e. with the change in the scenarios or situations, algorithms known to perform better in that situation are implemented and when the scenario changes, different algorithm can be used to fulfil the needs.

The decision of load balancing and scheduling can be made centrally i.e., by a single designated device or distributed i.e., by all the devices in a system. Decision taken by a single system can save time and resources but can easily become bottleneck for the entire system, whereas decision taken by group of nodes may take better decision but at the cost of coordination time.

Various algorithms for load balancing and scheduling are available in cloud computing technology [11] which after some modifications can be applied to fog computing environment since as compared to cloud, resources are limited, devices are heterogeneous in nature and tasks are latency sensitive.

3 Exploratory Study on Load Balancing and Scheduling Algorithms

Researchers addressed the issues of scheduling, load balancing, task allocation or resource allocation in the fog integrated cloud environment in many research papers given below and most of them addressed the centralized scheduling, load balancing and offloading of tasks. In centralized version of the problem, either entire information about the tasks, network, or nodes is known previously, or a centralized system decides task should be allocated to which device. In the distributed version of the problem there is no central device responsible for allocation of tasks and not even full information about the tasks, devices or network is known.

Ningning et al. [12] proposed a task allocation technique based on graph partitioning in fog computing environment. Here, depending upon the level of resources required, tasks can be assigned to a single or multiple virtual machines. Physical nodes of the fog computing layer are represented here as a non-directional graph. A minimum spanning tree is constructed from the graph obtained and edges that do not provide enough resources are removed. The tree obtained hence represents partition that can be used for task allocation. It is demonstrated that it leads to improvement in tasks' run time. Limitation of this approach is that frequent graph repartitioning is required when there is change in fog devices.

Salim Bitama et al. [13] proposed Bees Life Algorithm (BLA) which is a bio-inspired optimizing approach for job scheduling problem in the fog computing environment. An architecture is proposed in which user when sends a request to a nearby fog, it forwards data and parameters of that request to administrator node. The administrator node will decompose the job into a set of tasks and executes BLA to find the node which best executes the tasks and forwards the task to the corresponding node. Administrator node after getting partial results from the devices prepares final and forwards it to the user. It is demonstrated that the algorithm requires less memory and lesser execution time than PSO (particle swarm optimization) and genetic algorithm.

TejaswiniChoudhari et al. [14] proposed priority based task scheduling algorithm in the fog integrated cloud architecture. The algorithm implemented first checks the availability of desired resources in fog servers to satisfy user's requirement and if resources are not available, the request is forwarded to the cloud layer. Here three queues of different priority level maintained- high, medium and low. According to the maximum allowed delay and the priority given by user, priority is given to a task. It is shown that priority levels when added to Optimize Response Time (ORT) and Reconfigure Dynamically (RD) algorithm in cloud only environment reduces cost but increases response time. It is also tested that when priority is added to efficient resource allocation (ERA) algorithm in fog cloud integrated environment, it reduces both response time and cost. However major drawback in this paper is that it considers static priority levels and does not consider classical problem of aging with priorities.

LindongLiu et al. [15] used Classification Mining technique for scheduling of task. I-Apriori algorithm is introduced after applying some modification on Apriori algorithm and using it association rules are generated. Rules generated are input to TSFC algorithm (Task Scheduling in Fog Computing) which schedules tasks to execute on fog devices so as to minimize the completion time. It is shown that TSFC gives reduced execution and waiting time than MCT, MET, and MIN-MIN but it does not consider bandwidth between processors.

JinlaiXu et al. [16] proposed Zenith, a model for allocation of resources. This model can be used for establishing contracts between service providers and infrastructure providers for edge layer, for allocation of resources so as to maximize profit for both and to reduce latency for user. Allocation of resources here is decided by using auction mechanism in which firstly, utility function for both the service providers and infrastructure providers is identified, and then strategy for bidding is given. Finally an algorithm is introduced to decide selection of winner. Performance of this model when compared with Coupled edge computing (CEC) and conventional cloud model gives better response time for any number of servers in edge layer and better utilization and success rate when number of server increases.

Changlong Li et al. [17] proposed SSLB (self-similarity-based load balancing) technique, for fog servers spread over large scale by using both centralized and decentralized mechanisms. In this, k-means clustering algorithm is used to group fog nodes into cells. Scheduling is done independently in each cell. Workload migration across cells is done based on two adaptive threshold policy called Task Distributing and Task Grasping. It is further shown that utilization of resources in SSLB is better than DS (based on Karger's work) and more stable than CS (based on Apache Hadoop). It is also shown that with increasing number of fog nodes SSLB performs better than centralized scheme and better than decentralized scheme with reduced network bandwidth.

Xuan-Quy Pham and et al. [18] proposed a distributed task scheduling algorithm CMaS (Cost-Makespan aware Scheduling) in which fog computing works in collaboration with cloud computing. Its objective is minimize both execution time of application and the cost of resources. It consists of three phases- task prioritizing based on upward ranking, node selection phase to assign task to a fog or cloud node based on utility function, and task reassignment phase to meet user defined deadlines. It is also shown that CMaS performs better

than GfC (greedy for cost) in terms of performance and better than HEFT and CCSH in terms of cost.

Guangshun Li et al. [19] proposed a resource scheduling algorithm called FCAP which uses FCM (fuzzy C-Means clustering) to find cluster of fog resources and PSO (particle swarm optimization) to obtain global optimization which reduces scale of resource search and then weighted matching method is used to match the user request and the scheduling scheme is returned. It is shown that FCAP converges faster than FCM algorithm. It is also shown that in comparison with the Min-min algorithm, this algorithm can lead to better user satisfaction.

Sabihe Kabirzadeh et al. [20] proposed hyper-heuristic algorithm for scheduling in fog environment. Testing and selection technique is used to identify best algorithm for the new workflow among the available algorithms. In training phase, for initial set of workflows, different heuristic algorithms GA, PSO, ACO, and SA are implemented to allocate resources. After calculation of energy consumed, total cost and usage of network, it is stored in a database. In the test phase, for each new input topology, a row of training dataset having least Euclidean distance with input is selected and algorithm best for that topology is chosen (mentioned in database table). It is further shown that it consumes less energy than other heuristic algorithms and average cost is improved.

Sagar Verma et al. [21] proposed load balancing algorithm based on replication technique for a fog integrated cloud environment. By maintaining replicated data in fog networks, dependency on cloud data centers will reduce leading to reduced latency time. The request from the client when reaches the nearest fog server and if it is not capable of providing the data and resources itself the request is forwarded to the adjacent server in fog tier for handling the request. If the fog tier server fails to provide data or services requested by the user, the request is migrated to the cloud tier. Cloud server will then process the request if the request data is available there. If the cloud server also cannot serve the request due to un-availability of data or resources, it broadcasts the query packet throughout the Cloud tier. It is shown that this algorithm gives reduced response time than Round Robin and Throttled but high cost due to data transfer.

Fatima et al. [22] proposed dynamic service broker policy (DSP) in fog integrated cloud environment for energy management in residential buildings in all regions of the world. Depending upon the allocation of VMs, the fog device which is able to manage load with minimum delay is selected, otherwise request is forwarded to cloud. It is shown that the DSP policy when compared to dynamic configure with load (DR) policy for RR algorithm, gives better overall response time and processing time but the cost of virtual machines increases.

Maria et al. [23] proposed integration of cloud with smart grid for proper utilization of energy by users. Here performance maximization is mathematically formulated and performance of round robin (RR), throttled and weighted round robin (WRR) algorithms is analysed with various service broker policies under cloud fog environment. It is shown that Throttled algorithm performs better than RR and WRR.

Saman Zahoor et al. [24] used PSO algorithm in cloud-fog environment which can give globally best solution in addition to locally best. For each iteration, current local best value or global best value of previous iteration, whichever meets the fitness criteria, gets selected for

current iteration. Simulations are done to show that PSO minimizes response time and cost as compared to RR and Throttled.

Ye Yu et al. [25] proposed SDLB (scalable and dynamic load balancer) for fog and MEC. Here POG data structure is used to develop SDLB, which makes use of minimal perfect hashing technique. It is also shown that SDLB when compared with other load balancer design making use of hash table and consistent hashing is faster and consumes less memory.

Huynh et al [26] proposed TCaS algorithm based on genetic algorithm to allocate bag of tasks in cloud integrated fog environment to minimize both execution time and cost of operations. Here, a utility function is formulated to achieve minimize makespan and cost. Genetic algorithm is used with two-point crossover operation and Darwin's law of survival is used as selection strategy. Simulations are done to show that TCaS performs better than BLA for task completion time and achieves better fitness value of cost and makespan trade-off.

Deng et al. [27] tried to achieve balance between transmission delay and power consumption in the fog integrated cloud environment. In this paper workload allocation problem is formulated in which minimizing the power consumption with fulfilling delay constraint is divided into three sub-problems- tradeoff between power consumption and delay in fog layer, cloud layer and minimization of communication delay for dispatching, which can be solved within corresponding sub-systems. Hungarian method is used to solve delay minimization sub problem. Simulations shows that with increase in workload, power consumed and processing delay increases for fog only systems, power increases but delay constant in cloud only system and power increases and delay decreases in fog integrated cloud system.

Lin Gu et al. [28] investigated association of user with a base station, distribution of tasks and placement of VM in order to minimize cost in medical cyber-physical systems. The problem is firstly formulated into a non-linear linear program of mixed-integer (MINLP), which is then linearized into mixed-integer linear programming (MILP). Heuristic algorithm based on two-phase linear programming is proposed to solve the sub problems. Results of this algorithm is compared with greedy algorithm to shows that 2 phase LP gives minimum incremental cost which includes both VM deployment cost and communication cost between base stations.

Oueis et al [29] proposed a heuristic algorithm for resources management for clustering of fog servers which supports customizable objective like minimizing latency or minimizing power consumption, etc in the small cell cloud. In this paper, establishment of clusters and allocation of resources is split into two phases- First, computational resources are allocated locally to small cell. Secondly, clusters (SCC) are established for user left in first step due to scarce resources. To reduce complexity of problem, optimal cluster is found for each user, independently of the others. Three algorithms proposed are- EDF-PC (Earliest Deadline First with minimizing power consumption), EDF-LAT (Earliest Deadline First with minimizing latency), and CS-LAT (computation size with minimizing latency). The result shows that EDF-LAT and CS-LAT gives better user satisfaction than static and no clustering. Also they achieved significant latency gain compared to others and EDF-PC consumes lesser power than other two proposed techniques.

Xueying Guo et al. [30] proposed a task assignment policy in Edge-cloud system to reduce both power consumption and delay. Firstly the problem is modelled as a continuous-time queuing system and an equivalent discrete-time Markov decision problem is proposed to make the problem tractable. To reduce the complexity, an associated dual problem is analysed. Finally, an Indexing policy is proposed to handle the complex problem and reduce the overhead of communication. It is used to analyze the dual of the original problem in which the index is calculated based on their queuing status based on which users forward requests to cloud or edge servers.

Yousefpour et al. [31] proposed and formulated a cooperative offloading policy to forward requests from IoT devices to neighbouring fog nodes (Fog-to-Fog) or to forward them to the cloud (IoT-Cloud or Fog-Cloud) for delay minimization. If the estimated waiting time on a fog node is smaller than a threshold value, the task is accepted; otherwise, it is offloaded. There is a limit on the number of times a task can be forwarded to a neighbouring node, after which it must be forwarded to the cloud. It is shown through simulations that AFP (proposed policy-IoT process or forwards to cloud or fog) outperforms LFP (Light fog processing-only light request forwarded to fog or else to cloud if not processed by IoT) and NFP (no fog processing-IoT if not processed can only forward tasks to cloud). AFP also gives lesser delay than the Indexing policy [30]. A table showing the comparative study of various algorithms is given below.

Table 1 Comparison of Different Scheduling and Load Balancing Algorithms

Author(s) and Year	Main ideas	Simulation tool / Programming Language	Network type	Improved criteria	Limitation
Ningning et al. 2016 [12]	Task allocation based graph partitioning Constructed Minimum spanning tree	Extended Hadoop	Fog	Task's run time	Frequent graph repartitioning needed for dynamic load
Salim et al. 2017 [13]	Central node decomposes jobs into tasks. Bee Life Algorithm (BLA) is used to allocate task to nodes	C++	Fog	Reduced execution time and memory than PSO and genetic algorithm	Administrator node is central, hence may become bottleneck or single point of failure. Small dataset used
Tejaswini et al. 2018 [14]	Priority assigned to task based on given priority and max delay allowed.	Cloud Analyst	Cloud -Fog	Reduced the response time and cost as compared to ORT and RD algorithm.	Considers static priority levels. May suffer from problem of aging.
LindongLiu et al. 2018 [15]	Based on Classification Mining. Used I-Apriori (modified Apriori)	SimGrid	Fog	Reduced execution and waiting time than MCT, MET, and MIN-MIN algorithm.	Bandwidth between processors not considered

	Introduced TSFC algorithm				
Jinlai Xu et al. 2017 [16]	Based on an auction-based mechanism for resource allocation. Decoupled service and infrastructure management.	geographic map	Edge	Reduced response time and Increased utilization	Low utilization with scarce resources Assumes every container requires equal resources
Changlong Li et al. 2018 [17]	Uses both centralized and distributed technique k-means clustering used to group nodes. To avoid communication overhead, adaptive threshold policy is used	iFogSim	Fog	Resource utilization of SSLB better than traditional centralized and decentralized techniques. Performs better for large scale clusters.	Overhead of clustering. Network load due to probing and communication.
Xuan-Qui Pham and et al. 2017 [18]	Based on utility function that considers both cost of resources and finishing time of task. Performs task reassignment for meeting deadlines.	CloudSim	Cloud-Fog	Achieves trade-off between cost of resources and performance. Reduced cost than HEFT and CCSH Improved performance than GfC	CMaS cannot guarantee finishing of tasks before deadline. May not satisfy deadline for GE program
Guangshun Li and et al. 2019 [19]	Based on FCM and PSO algorithm. Creates cluster of fog resources to reduce search and uses weighted matching method to allocate resources.	MATLAB	Fog	Better user satisfaction than min-min algorithm Faster convergence than FCM algorithm	One task can be handled at a time Does not consider dynamic nature of resources.
Sabihe Kabi rzadeh et al. 2017 [20]	Hyper-heuristic algorithm based on classification technique. Consists of training and testing phases.	iFogSim	Fog	Reduced energy consumption and simulation time than other heuristic algorithms Improved decision making according to workflow	Initial training phase is an overhead and involves delay is start-up time.
Sagar Verma et al. 2016 [21]	Uses data replication technique. Proposed Cloud-Fog architecture.	Cloud Analyst	Cloud-Fog	Reduced response time than RR and Throttled. Reduced processing time on data center Reduced dependency on cloud layer	Increases total cost due to data transfer for replication.

Fatima et al. 2018 [22]	New service broker technique introduced.	Cloud Analyst	Cloud-Fog	reduced processing response time	cost of VMs increased Considered virtual machines with similar capabilities
Maria Naeem et al. 2019 [23]	Implemented Weighted round robin algorithm in fog-cloud environment. Mathematically formulated performance objective.	Cloud Analyst	Cloud-Fog	Considers different load distribution for different VM capabilities, hence balances load	Response time increases as compared to throttled algorithm.
Saman Zahoor et al. 2018 [24]	Implemented PSO in fog-cloud for resource management in SG	Cloud Analyst	Cloud-Fog	Reduces response time and cost as compared to RR and Throttled	It is specific for two buildings
Ye Yu et al. 2017 [25]	Proposed SLDB (Scalable and Dynamic Load Balancer). POG data structure (minimal perfect hashing)	HashMap using a self-balancing binary search tree	MEC-Fog	Less memory utilization Faster and better data plane throughput	Structures grows linearly with number of stateful IDs Time to rebalance increases with no. of packets
Huynh et al. 2018 [26]	Utility function to achieve trade-off between costs and makespan. Genetic algorithm with two-point crossover is used.	iFogSim	Cloud-Fog	Less time to complete tasks than BLA Better fitness value than BLA	Cost of TCaaS is more than BLA. Tasks are assumed to be independent.
Deng et al. 2016 [27]	Workload allocation is divided into three sub-problems Hungarian method is used	MATLAB	Cloud-Fog	Power consumption and delay	Implemented in centralized system
Lin Gu et al. 2015 [28]	Developed two-phase linear programming based on heuristic	Mathematical Gurobi	Fog	Reduced deployment and communication cost	Does not provide computational offloading capability.
Oueis et al. 2015 [29]	Resource allocation and cluster establishment using EDF-PC, EDF-LAT, CS-LAT. Heuristic	3GPP framework	Fog	Improved user satisfaction. EDF-PC is power efficient. Customizable algorithm for multi user scenario	Complex for large scale fog infrastructure
Xueying Guo et al. 2016 [30]	Used Markov decision problems for online task allocation. Index policy is proposed.	theoretical	Cloud-Fog	Reduced power consumption. Reduced delay	Continuous-time queuing system

Yousefpour et al. 2018 [31]	Task off-loading collaboration scheme. Offloading decisions based on queuing status	Not Mentioned	Cloud-Fog	Minimization of delay	Threshold parameter is kept static
------------------------------------	---	---------------	-----------	-----------------------	------------------------------------

Conclusion

In this paper, we have surveyed fog integrated cloud environment and its 3-layered architecture. We have identified the need for the development of new algorithms for scheduling and load balancing for fog integrated cloud computing keeping in view to reducing the latency time. This paper also gives an exploratory comparative study of the existing algorithms proposed by various researchers and also set grounds for further improvements and development of these algorithms. Since selecting the best fog server (or edge server) in order to minimize latency for every end device and maximizing resource utilization is NP-Hard problem, researchers are using various optimization techniques like graph theory, heuristics, local search, stochastic search, genetic algorithms, deep learning. Pros and cons of all the algorithms were discussed and the whole summary is placed in a tabular form. As fog computing technology is in its infant stage, there is lot of scope for improvement of the existing algorithms or defining new ones.

References

- [1]. Dave Evans. The internet of things: How the next evolution of the internet is changing everything. CISCO whitepaper, 2011.
- [2]. Cisco Global Cloud Index. Forecast and Methodology, 2016–2021. CISCO whitepaper. 2018
- [3]. Cisco Visual Networking Index. Forecast and Trends, 2017–2022 White Paper. Updated: February 27, 2019.
- [4]. Open Fog Consortium. Open fog reference architecture for fog computing. Feb 2017. [Online]. Available: <https://www.openfogconsortium.org/ra>.
- [5]. Subhadeep Sarkar, Sudip Misra. Assessment of the Suitability of Fog Computing in the Context of Internet of Things. IEEE. 2015.
- [6]. IEEE Standard Association. Ieee 1934-2018 - IEEE standard for adoption of openfog reference architecture for fog computing. [Online]. Available: <https://standards.ieee.org/standard/1934-2018.html>
- [7]. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog Computing and Its Role in the Internet of Things. Proceedings of ACM; MCC pp. 13–16. 2012.
- [8]. Ivan Stojmenovic, Sheng Wen. The Fog Computing Paradigm: Scenarios and Security Issues. Proceedings of Federated Conference on Computer Science and Information Systems, 2014. IEEE; 2014.
- [9]. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog Computing and Its Role in the Internet of Things. Proceedings of ACM; MCC pp. 13–16. 2012
- [10]. Blesson Varghese, Nan Wang, Sakil Barbhuiya. Challenges and Opportunities in Edge Computing. arXiv:1609.01967v1 [cs.DC] 7 Sep 2016.
- [11]. Subhadra Bose Shaw, Dr. A.K. Singh. A Survey on Scheduling and Load Balancing Techniques in Cloud Computing Environment. IEEE; 2014
- [12]. SONG Ningning¹, GONG Chao, AN Xingshuo, ZHAN Qiang. Fog Computing Dynamic Load balancing mechanism based on Graph Repartitioning. China Communications, March 2016.
- [13]. Salim Bitam, Sherali Zeadally, Abdelhamid Mellouk. Fog computing job scheduling optimization based on bees swarm. Published online: 7 Informa UK Limited, Taylor & Francis Group. 10 Apr 2017.

- [14]. Tejaswini Choudhari, Melody Moh, Teng-Sheng Moh. Prioritized Task Scheduling in Fog Computing. USA, ACM ; 2018
- [15]. LindongLiu, DeyuQi, NaqinZhou, andYilinWu. A Task Scheduling Algorithm Based on Classification Mining in Fog Computing Environment. Wireless Communications and Mobile Computing Volume, Wiley, Hindawi; 2018.
- [16]. JinlaiXu, BalajiPalanisamy, Heiko Ludwig and Qingyang Wang. Zenith: Utility-aware Resource Allocation for Edge Computing. IEEE International conference on Edge computing ,2017.
- [17].Changlong Li,Hang Zhuang,Qingfeng Wang,Xuehai Zhou. SSLB: Self-Similarity-Based Load Balancing for Large-Scale Fog Computing. Arabian Journal for Science and Engineering, December 2018, Volume 43, Issue 12, pp 7487–7498.
- [18]. Xuan-Quy Pham, Nguyen Doan Man, Nguyen Dao Tan Tri, Ngo Quang Thai and Eui-Nam Huh. A cost- and performance-effective approach for task scheduling based on collaboration between cloud and fog computing. International Journal of Distributed Sensor Networks. 2017. Vol. 13(11).
- [19]. Guangshun Li , Yuncui Liu, Junhua Wu , Dandan Lin andShuaishuai Zhao. Methods of Resource Scheduling Based on Optimized Fuzzy Clustering in Fog Computing. Sensors , MDPI;2019.
- [20]. SabiheKabirzadeh, DadmehrRahbari, Mohsen Nickray. A Hyper Heuristic Algorithm for Scheduling of Fog Networks. proceeding of the 21st conference of fruct association, IEEE ;2017.
- [21]. SagarVerma, Arun Kumar Yadav, Deepak Motwani. An Efficient Data Replication and Load Balancing Technique for Fog Computing Environment. Proceeding of 3rd International Conference on Computing for Sustainable Global Development, IEEE; Oct 2016.
- [22]. Itrat Fatima, NadeemJavaid, Muhammad Nadeem Iqbal, IsraShafi, Ayesha Anjum, UbedUllah Memo. Integration of Cloud and Fog based Environment for Effective Resource Distribution in Smart Buildings. 14th International Wireless Communications & Mobile Computing Conference (IWCMC).IEEE; Aug 2018.
- [23]. Maria Naeem, NadeemJavaid, MaheenZahid, Amna Abbas, Sadia Rasheed, and SaniahRehman. Cloud and Fog Based Smart Grid Environment for Efficient Energy Management. INCoS 2018, Springer 2019, LNDECT 23. pp. 514–525.
- [24]. SamanZahoor, NadeemJavaid, Asif Khan, BibiRuqia Fatima J. Muhammad, Maida Zahid. A Cloud-Fog-Based Smart Grid Model for Efficient Resource Utilization. 14th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE; August 2018.
- [25]. Ye Yu, Xin Li, and Chen Qian. Sdlb.:A scalable and dynamic software load balancer for fog and mobile edge computing.Proceedings ofthe Workshop on Mobile Edge Communications. ACM 2017. pages 55–60.
- [26]. Huynh ThiThanhBinh, Tran The Anh, Do Bao Son, Pham AnhDuc, Binh Minh Nguyen. An Evolutionary Algorithm for Solving Task SchedulingProblem in Cloud-Fog Computing Environment. International Symposium on Information and Communication Technology ,SoICT. ACM; 2018.
- [27].Ruilong Deng, Rongxing Lu, ChengzheLai, Tom H. Luan,andHaoLiang. Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption. IEEE Internet of Things Journal. Dec 2016;Volume: 3 Issue: 6.
- [28]. Lin Gu, Deze Zeng, Song Guo, Ahmed Barnawi, and Yong Xiang. Cost Efficient Resource Management in Fog Computing Supported Medical Cyber-Physical System. IEEE Transactions on Emerging Topics in Computing. December 2015; Volume: 5, Issue: 1,Jan.-March 2017.
- [29]. Jessica Oueis, Emilio Calvanese Strinati, and Sergio Barbarossa. The Fog Balancing: Load Distribution for Small Cell Cloud Computing. IEEE 81st Vehicular Technology Conference (VTC Spring); 2015.
- [30]. Xueying Guo, Rahul Singh, Tianchu Zhao and Zhisheng Niu. An Index Based Task Assignment Policy for Achieving Optimal Power-Delay Tradeoff in Edge Cloud Systems. IEEE International Conference on Communications (ICC); 2016.
- [31]. Ashkan Yousefpour, Genya Ishigaki, Riti Gour, and Jason P. Jue. On Reducing IoT Service Delay via Fog Offloading. IEEE Internet of Things Journal. April 2018; Volume: 5 Issue: 2.