Resource Management Strategies in Fog Computing Environment –A Comprehensive Review

Deafallah Alsadie
dbsadie@uqu.edu.sa
Department of Information Systems
Umm Al-Qura University
Makkah, Saudi Arabia

Abstract: Internet of things (IoT) has emerged as the most popular technique that facilitates enhancing humans' quality of life. However, most time sensitive IoT applications require quick response time. So, processing these IoT applications in cloud servers may not be effective. Therefore, fog computing has emerged as a promising solution that addresses the problem of managing large data bandwidth requirements of devices and quick response time. This technology has resulted in processing a large amount of data near the data source compared to the cloud. However, efficient management of computing resources involving balancing workload, allocating resources, provisioning resources, and scheduling tasks is one primary consideration for effective computing-based solutions, specifically for time-sensitive applications. This paper provides a comprehensive review of the source management strategies considering resource limitations, heterogeneity, unpredicted traffic in the fog computing environment. It presents recent developments in the resource management field of the fog computing environment. It also presents significant management issues such as resource allocation, resource provisioning, resource scheduling, task offloading, etc. Related studies are compared indifferent mentions to provide promising directions of future research by fellow researchers in the field.
Keywords: *Fog computing, Resource management, Task scheduling, Resource provisioning.*

1. Introduction

Internet of things (IoT) technology facilitated connecting everyday devices with the Internet, leading to helpful interaction between machines and humans. It connected different sensors, actuators, and device controllers [1]. It has become popular in different applications such as health care, retail, Industrial Automation transport etc. IoT applications can produce a large amount of data using different mobile devices and sensors, leading to issues with traditional cloud computing environments such as network bandwidth, latency and security issues [2, 3].

In order to address the issues of traditional cloud computing, a new distributed computing Technology called for technology have been proposed. Fog technology acts as a middleware between IoT devices and cloud servers and helps meet the computational needs of letters in sensitive applications [4, 5].

Fog computing environment contains processing elements, network devices and storage devices so that computational services get closer to the IoT devices. It helps to
reduce energy consumption, network bandwidth and latency compared to direct cloud communication [6]. However, fog nodes are generally capacity constrained regarding processing and storage capabilities. So these nodes cannot be considered dedicated servers. In addition, computing nodes have power consumption issues, efficient resource allocation, problems with handling the fluctuating workload. These issues require effective management of computing resources in the fog computing environment.

Therefore, this paper targets a comprehensive review of computing Resource Management in the fog computing environment considering different aspects such as application placement, resource scheduling, computing resource allocating, task offloading, workload balancing. This paper contributes in the following ways.

- It presents a comprehensive review of resource management strategies in the fog computing environment.
- It highlights different critical issues of resource management strategies in the fog computing environment.
- It provides a comprehensive comparison of different resource management strategies

The rest of the paper is structured as follows. Section 2 describes related reviews in resource management in the fog computing environment. Related studies are compared in different parameters. Section 3 presents fog computing architecture in brief. Different resource management strategies in the fog computing environment have been presented in section 4. Section 5 highlights critical issues related to resource management in the fog computing environment. Finally, the paper is concluded in section 6.

2. Related Work

This section describes and analyses studies related to fog and cloud computing resource management. It provides a comprehensive comparison of different studies in multiple dimensions. Mouradian et al. [7] presented an analysis of fog computing approaches regarding their architecture and algorithms. They provided critical research challenges and open issues along with the research directions in their review.

Manuscript received April 5, 2022
Manuscript revised April 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.4.38

However, the authors have not considered many algorithmic dimensions resource provisioning parameters. They only described fog computing approaches for managing resources based upon a few algorithmic metrics.

Hong et al. [8] presented a review of various technical issues to manage resource-limited fog and edge computing environments. Their identified issues regarding infrastructure, algorithms and architecture. They presented four algorithms related to discovery, load balancing, placement, and benchmarking from an algorithm perspective. In terms of infrastructure, they provided three categories of resources such as system software, middleware and Hardware. They presented division of architecture as data flow, tenancy and control in their review. However, their review has not considered some aspects of resource management in fog computing, such as resource provisioning and resource allocating.

Aazam et al. [9] reviewed various task offloading methods in the fog and edge computing area. They suggested the taxonomy off-task offloading method, highlighted research issues and provided future research directions in the offloading task field. They have also identified Different IoT middleware techniques such as mobile edge computing, cloudlet and micro data centre to enable different Technologies such as virtualization and wireless communication. They have also divided offloading criteria in the fog computing environment, latency, energy consumption, load balancing. Their study only focused on task offloading methods. They ignored other resource management methods such as resource allocating, resource provisioning, load balancing and scheduling.

Masip-Bruin et al. [10] reviewed different resource management issues in edge and cloud computing. Accordingly, they presented a layered architecture for resource provisioning and efficient resource selecting and service exhibition methods. They have also described performance benefits regarding database size based upon layered architecture. They also presented future research directions in this area. The authors mainly focused on architectural aspects. But they have ignored the algorithmic aspect in managing resources of the fog computing environment. They did not present any issues related to resource management in the fog computing environment regarding load balancing, task offloading and application placement methods. The authors of [11] presented a taxonomy of resource management methods in an edge computing environment consisting of four categories, resource type, resource management goal, resource usage and location.

Dias de Assunção et al. [12] presented a survey of stream processing methods to extract resource elasticity features in the edge computing environment. In this work, the authors mainly focused on auto scaling solutions for dynamic resource provisioning and resource elasticity in context of resource management in the fog computing environment. They listed out ongoing effects on resource elasticity and their deployment in the edge computing environment. The main focus of research in this work is resource management challenges in the edge computing environment. However, they considered non-functional metrics limited for resource management methods in the fog computing environment.

Naha et al. [14] presented a survey on resource allocation and scheduling methods in the fog computing area. They summarized recent studies by focusing on resource allocation in the fog computing environment. However, they have not considered load balancing, quality of service and energy efficiency in their survey.

Bendechache et al. [15] reviewed recent work on resource allocation of the fog computing environment. They provided a classification of resource management metrics, resource scheduling and resource provisioning. They categorized resource provisioning metrics into three classes, selection, detection and mapping. In contrast, they categorize resource scheduling metrics as monitoring, allocating and load balancing. In addition, they have also reviewed different key performance indicators for computing environments such as latency, Virtual Machine placement, scalability, failure rate, resource use usage, power consumption, efficiency, cost and service level agreement violations.

Salaht et al. [17] analyzed optimisation metrics for addressing resource management and service placement problems in the fog computing environment. The authors focused on different metrics in evolving latency, cost, resource usage, energy consumption, quality of experience injection rate and blocking chances. The authors have also provided promising research directions by highlighting different research issues in the server: replacement problems, optimisation method, and evaluation frameworks.

Ghobaei-Arani et al. [1] presented a taxonomy of resource management methods for cloud computing environments by considering six dimensions, resource scheduling, application placement, resource provisioning, task offloading, resource allocator and load balancing. They provided a comprehensive review of different case studies, their methods, performance metrics, and different tools by highlighting the pros and cons of a prospective study.

The authors of [16] presented taxonomy for different types of metrics regarding performance cloud models and o MAPE-K concept. They divided standard performance metrics in the computing area into seven metrics, network congestion, throughput, statistical analysis metrics, scalability, profitability, fault tolerance and service level agreement violation. In the context of cloud models, they classified metrics into six categories, public, private, hybrid, single service provider, multiple service provider and federation. They have also highlighted four categories of metrics as per the MAPE-K loop, analysing, monitoring, executing and planning.

Table 1 provides a comprehensive comparison of above mentioned studies in different dimensions. It can be concluded that most of the studies have not categorised resource management methods in fog computing in a reasonable way. Unorganised review studies of resource management methods in limited dimensions is the primary motivation for conducting a comprehensive review of resource management methods in fog computing in this work.

3. Fog computing setup

Figure 1 presents a high level layered architecture of the fog computing environment. For computing environment consists of three layers, corresponding to IoT devices / sensor, for computing layer and Cloud Computing layer [19].

The bottom layer comprises many Internet connected devices and sensors. The fog layer collects data generated by different Internet connected devices and sensors through for gateways. The fog layer contains some processing capability to minimise execution time and bandwidth at cloud data centres which are far away from the bottom layer [20]. The fog layer act as a middle layer containing many fog nodes. The fog nodes are connected to the cloud through cloud gateways and transmit workload to the cloud server after fixed intervals. The topmost layer contains cloud data centres for the processing and storing of data generated by Internet connected devices and sensors [21].

Table 1. A comparison of resource management method's reviews in fog computing

Study	Main focus	Architecture	Application placement	Resource scheduling	Task offloading	Load balancing	Resource allocation	Resource provisioning	QoS
Mouradian et al. [7]	• Architecture • Algorithms	Yes	No			No		No	
Aazam et al. [9]	• Task offloading techniques taxonomy, middleware technologies	No	No	No	Yes	No	No	No	No
Hong and Varghese [13]	• Technical challenges for managing the resource-limited in fog and edge computing in terms of Infrastructure, Architectures and Algorithms	Yes	Yes	Yes	Yes	Yes	No	No	No
Masip-Bruin et al. [10]	• Resource management of edge to the cloud computing • Layered architecture for resource continuity provisioning for service execution and resources selection methods	No	No	Yes	No	No	Yes	Yes	No
Neha et al. [14]	• Evaluation framework for resource management techniques	No	No	No	No	No	Yes	Yes	No
Bendechache et al. [15]	• Resource allocation, Resource scheduling, and Resource provisioning	No	No	No	No	No	Yes	Yes	No
Mostafa Ghobaei-Arani [1]	• Resource management approaches in fog Environment	No	Yes	Yes	Yes	Yes	Yes	Yes	No
Aslanpour et al. [16]	• Taxonomies for evaluating cloud, fog, and edge computing • Analysis of performance metrics • Analysis of cloud model metrics								
Salaht et al. [17]	• Optimization metrics for Resource management and service placement problems • Summary of metrics								

4. Resource Management in Fog Computing

Many researchers have discussed various resource management strategies in fog computing based upon different criteria such as resource allocation, work load balancing, resource provisioning, resource scheduling etc. The main objective of resource management strategies to reduce power consumption, communication cost and latency in the fog computing environment. Several metrics have been proposed to measure the performance of different resource management algorithms under different evaluation frameworks. Evaluation metrics depends upon different deployment scenario, criteria adopted for resource management, power management and quality of service [22]. The authors of [23] presented a resource

Manuscript received April 5, 2022
Manuscript revised April 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.4.38

management method in the fog architecture layer. They focused on optimizing reliability and latency in the fog computing environment. They proposed a consumer layer for completing specific demands through fog and cloud

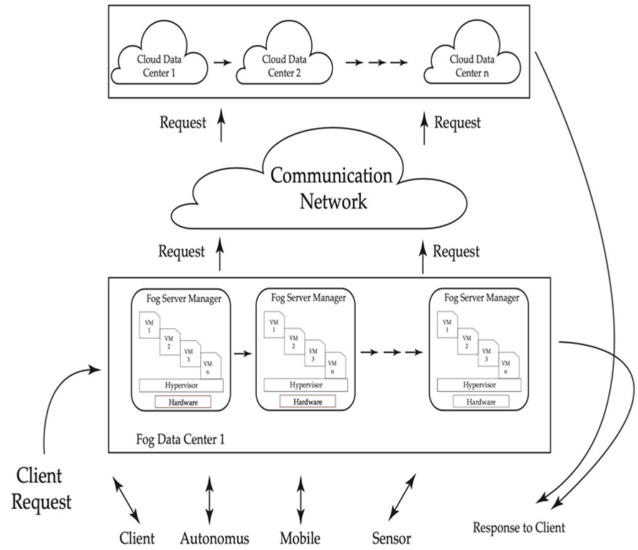

Fig. 1 layered architecture [22]

The authors of [24] proposed a dynamic resource allocation strategy based on dynamic resource scheduling and static resource allocation to achieve dynamic load balancing in the fog computing environment. Agarwal et al. [18] proposed an efficient resource allocation approach for maximizing throughput and minimising response time in for computing environment. Similarly, Taneja and Davy [25] focus on minimising energy consumption and latency using an iterative algorithm.

This work reviews and summarizes findings of recent resource management strategies as per taxonomy presented in Table 2 [1] in the following subsections.

4.1 Application Placement Methods

Minh et al. [31] focused on optimising service decentralization in fog computing based upon multilayer architecture. They considered criteria of IoT service number and IoT services for deploying IoT services in fog computing. Similarly, Saurez et al. [32] introduced a foglet model for geographically distributed computation. The proposed solution automatically helps discover resources and placement applications in the fog computing environment.

Brogi et al. [33] suggested a quality of service aware method for deploying IoT applications in for computing computing. They focused on request per hour, processing time and response time based **upon** the Round Robin strategy in their algorithm.

Several researchers have focused on application Placement in the fog computing environment and proposed different frameworks. For example, Skarlat et al. [26] presented a service-based placement method for computing environments considering application quality of service criteria and resource heterogeneity. They used a genetic algorithm based approach for finding appropriate service placement in the fog computing environment. Experimentally validated that their proposed approach is effective in service placement compared to other criteria such as resource utilisation and application execution time.

Venticinque et al. [27] proposed a methodology for benchmarking, evaluating and testing IoT solutions. Mahmoud et al. [28] introduced an energy aware task and locating approach. This focus on enhancing energy consumption and latency factors. Yangui et al. [29] suggested a PaaS based model for automating the deployment of IoT applications in the fog computing environment based upon representational state transfer protocol.

Table 2. Taxonomy of resource management stratigies [1]

Virtual machine placement methods	Application placement,
	Load balancing,
	Resource allocation, and
	Resource provisioning
	Resource scheduling
	Task offloading

Yigitoglu et al. [30] develop a Framework for managing automated IoT application deployment. The major components of this framework include a version control server, container registry component, orchestration server, fog nodes, and tool for continuous integration.

environment. Yao et al. [34] analysed deployment of cloudlets servers of bearing capacity and user requirements without compromising the quality of service. They proposed their solution in two phases: heterogeneous cloudlet service selection and deployment. For cloud service selection, they proposed a greedy algorithm for minimising cost and for deployment, they used resource capacity end user mobility.

Yousefpour et al. [35] suggested a dynamic service provisioning solution for deploying IoT services in the fog computing environment dynamically while meeting the quality of service requirements regarding bandwidth utilisation and latency. They used integer nonlinear programming in their proposed approach to minimise service level agreement violations and reduce resource costs.

Taneja et al. [25] suggested a solution for deploying IoT applications in the fog computing environment based upon different algorithms. Their approach maps the placement of IoT applications on network nodes search for appropriate nodes to meet application requirements. They demonstrated that their solution is generic and applies to a wide range of IoT applications in different topologies.

The authors of [36] suggested a latency sensitivity application method in distributed computing environment for addressing the issue of diverse service delivery latency of applications. They proposed their method based upon algorithms related to the application placement module and application forwarding module. They demonstrated their approach based on that line satisfaction ratio and placement time compared to the conventional latency aware method.

Naranjo et al. [37] suggested a framework to manage smart city devices in fog computing. They focus on addressing scalability, energy consumption and latency requirement issues in fog computing. The authors of [38] introduced a quality of experience application replacement method using fuzzy logic for characterizing the various application request that helps to deploy applications in the fog computing environment.

Velasquez et al. [39] proposed a modular architecture for placing IoT services in the fog computing environment. They focused on providing an intelligent service placement approach to facilitate IoT location service as per application requirements. The proposed architecture contains three modules: service repository, information collecting module, and service orchestrator module. Service orchestrator module act as the brain of the architecture for implementing the methods to take appropriate decisions regarding service placement in the fog computing environment.

Selimi et al. [40] described a lightweight method for service placement in a community network. Their experimental results demonstrate the algorithm's complexity as a full algorithm and better response time and bandwidth performance.

The above mentioned application placement methods in for computing environment can be observed that there are mainly three types of solutions based on models, search and framework. Model based solutions mainly focus on mathematical models such as integer linear programming for optimising specific objective functions. Search Bay solutions have been proposed or based upon specific heuristic and metaheuristic methods such as a genetic algorithm. Framework based solutions provide a framework for solving application placement issue.

Table 3 presents a comprehensive summary of application placement based resource management strategies mentioned above.

4.2 Resource Scheduling Methods

Many researchers focused on resource scheduling methods for the effective scheduling of fog computing resources. In the fog computing environment, devices can request different services fulfilled by fog nodes. Each service request contains a set of tasks. Resource scheduling methods find an appropriate node for executing different tasks while meeting service requirements' quality and minimising execution time in the fog computing environment.

Bitam et al. [41] proposed a bio-inspired algorithm-based method for scheduling different jobs in the fog computing environment. They focused on optimising two criteria, memory allocated to services and execution time in the fog computing environment. For example, Sun et al. [42] suggested a bi level resource scheduling method. The first level determines the scheduling of resources indifferent for clusters followed by scheduling for the nodes in the same cluster to complete the task on its arrival. They used non-dominated sorting algorithm -II for reducing service latency and enhancing stability while executing the task.

De Benedetti et al. [43] developed a distributed task scheduling method. There proposed method apps to automate many task execution using lightweight message bus approach. Cardellini et al. [44] prevented a distributed quality of service oriented method for scheduling data stream processing systems. Their proposed solution consists of extended strom with additional components for monitoring incoming and outgoing data transfer rate on fog nodes and quality of service monitoring for approximating network latency.

Rahbari et al. [45] suggested a novel scheduling method using a knapsack algorithm for reducing energy consumption and delay in the computing environment.

Zeng et al. [46] suggested a three step Framework for task scheduling in fog computing for minimising the latency of the request. This approach addresses computation time and coupling input and output handling operations during task competition.

Pham et al. [47] used a heuristic based approach for scheduling tasks in the fog computing environment. In their approach, they determine task priority and then select the appropriate node on the basis of earliest start time and earlier finish time in fog computing.

Fan et al. [48] suggested a deadline oriented task scheduling method using a multidimensional 0/1 knapsack problem. They proposed using the ant Colony optimisation algorithm to improve overall profit in the cloud computing environment while satisfying resource constraints and task deadlines.

Sun et al. [49] suggested a task scheduling method using Game Theory and a crowdfunding approach. They focused on encouraging service owners to rent their underutilized resources. The underutilized resources are identified using resource assistants.

Chen et al. [50] introduced dynamic scheduling methods based upon response time and queue length differences info computing environment. Deng et al. [51] proposed an approximation solution for addressing the issue of power consumption delay trade off in the fog computing environment.

Table 3. Summary of application placement based resource management strategies

Study	Method employed	Performance metrics	Pros	Cons
Skarlat et al. [26]	Genetic algorithm based method	• Execution delay • Service placement • Execution cost	• Considers heterogeneity of applications and resources	• Ignored resource cost • Framework specific
Venticinque et al. [27]	Heuristic based method	• Execution time • CPU load • Network usage	• Improved communication overhead • Improved bandwidth • Good performance • High throughput	• Power consumption not validated • Ignored deploying of IoT application
Mahmoud et al. [28]	Heuristic based method	• Latency, power • Consumption, network • Usage	• Improved power consumption • Improved latency	• Ignored mobility aspect • High computational complexity
Yangui et al. [29]		• End-to-end delay	• Automatic deployment of applications • Improved end-to-end delay	• Ignored weighting factors • Power consumption not validated • Cost not validated
Yigitoglu et al. [30]		• Latency • Bandwidth	• Container based virtualization support • Resources heterogeneity	• Improved scalability • Not validated for real time IoT application
Minh et al. [31]	Heuristic based method	• Latency • Power usage • Network usage	• Context aware information • Improved latency • Power usage • Network usage	• Ignored resource cost • High computational overhead
Saurez et al. [32]		• Latency • Number of the fog node	• Model for situation awareness applications • Container based virtualization support	• Ignored power consumption • Ignored overhead
Brogi et al. [33]	Backtracking based method	• Design time • Deployment • Time • Run time	• Improved latency • Bandwidth usage • Supports application lifecycle	• High computational complexity • Power consumption has not validated
Yao et al. [34]	Heuristic based method	• Deployment cost	• Improved computational complexity • Considered heterogeneity of cloudlets • Considered mobility pattern	• Power consumption not validated • Not validated in a real world application
Yousefpour et al. [35]	Heuristic based method	• Service delay • Number of fog service • Cost	• Improved service latency • Improved computational complexity	• Power consumption has not validated
Mahmud et al. [36]	Heuristic based method	• Deployment time • Deadline • Number of the fog node	• Considered latency-aware IoT application • Improved deployment time	• Not validated for real world scenario • Not considered customized settings and mobility

Naranjo et al. [37]		• Latency • Power consumption	• Improved power consumption • Considerd communications between IoT devices	• Improved scalability • Ignored cost • Ignored real-time data processing
Mahmud et al. [38]	Fuzzy logic based method	• Application placement • Time • Processing time • Reduction ratio • Network relaxation ratio • Resource gain	• Improved the processing time • Improved application placement time • Considered expectations of the applications users • Considered fog resources	• Not validated in a real world scenario • Overhead not analyzed
Velasquez et al. [39]	Integer linear programming	• Latency • Hop count, • Number of service • Migrations	• Improved the service latency • Migration of IoT services	• Not validated power consumption • Lack of simulations •
Selimi et al. [40]	Heuristic based method	• Response time • bandwidt	• Improved computational complexity • Considered changing of network topology conditions	• Improved scalability • Not validated latency

Table 4. Summary of resource scheduling based resource management strategies

Hoang et al. [52] focused on addressing task scheduling optimisation while distributing tasks between local reasons and remote cloud servers, aiming minimization of task execution time in fog computing.

The above cited resource scheduling methods in the fog computing environment show that most researchers propose dynamic scheduling approaches for scheduling resources. They mainly focused on minimising latency, response time and increasing efficiency dynamically.

Table 4 presents a comprehensive summary of resource scheduling based resource management strategies mentioned above.

A. Task offloading methods

Task offloading methods handle computing resource constraints such as battery backup, power and storage space of IoT devices and sensors. Generally, mobile devices are resources limited that are deployed in the fog computing environment. For practical completion, a few tasks must be outsourced to fog or cloud processers to improve their performance and optimise battery consumption.

Several methods have been proposed to address the task offloading issue in fog computing. For example, Tran et al. [53] suggested offloading method as a service for addressing the issue of task offloading related to memory, CPU and battery backup in fog computing. They used matching theory in developing their approach. Liu et al. [54] analysed trade off among different offloading metric structures offloading payment cost, energy utilisation and delay. They solved the issues using multi-objective queue models by finding optimal loading possibilities in the mobile computing environment.

Mukherjee et al. [55] proposed a cooperative code offloading method in the mobile computing environment. They focused on recording tomato let and co-operate by contributing their resources. Wang et al. [56] proposed an offloading method based upon queueing theory to minimise the response time of events in the fog computing environment of vehicles. The authors of [57] proposed offloading solution using reinforcement learning to minimise overall cost in the mobile computing environment. Liu et al. [58] designed a socially aware framework for task offloading in fog computing. They focused on the energy consumption of social relationships among mobile devices. Their framework used game theory for

Study	Method employed	Performance metrics	Pros	Cons
Bitam et al. [41]	Algorithm Bees Life basedmethod	• Execution time • Allocated memory	• Improved execution time • Managed allocated memory	• Improved scalability • Static scheduling
Sun et al. [42]	NSGA-II based method	• Service latency • Stability	• High scalability • Improved latency • Improved execution time	• High cost
De Benedetti et al. [43]	Adaptive based method	• Scalability • Fault tolerance	• Improved latency • Improved execution time • High interaction with IoT • Devices	• Improved scalability • High cost
Cardellini et al. [44]	Adaptive based method	• Node usage • Application latency • Inter-node traffic	• Improved latency • Improved execution time • Enhancing runtime • Scheduling	• Centralized topology • Improved availability • Improved scalability
Rahbari et al. [45]	Search symbiotic organisms based method	• Power usage • Network usage • Cost	• Improved execution cost • Saving sensor lifetime • Optimized power usage	• High execution time
Zeng et al. [46]	Heuristic based method	• Power consumption	• Improved computational complexity • Improved power consumption	• Improved scalability • Not validated latency
Pham et al. [47]	Heuristic based method	• Cost makespan tradeoff	• Balancing cost • Makespan	• High workload execution time • Improved scalability
Fan et al. [48]	ACO based method	• Total profit • Guarantee ratio	• Optimized power usage • Optimized profits of fog providers	• High execution time • High time complexity
Sun et al. [49]	Game theory based method	• Completion time • Sla violation rate	• Improved -performance time • Improved the sla violation rate	• High cost • High power consumption
Chen et al. [50]	Heuristic based method	• Response time • Queue length	• High dynamic efficiency • Using a formal method • Improved time	• Not validated for complex scenarios
Deng et al. [51]	Approximation based method	• Power consumption • Latency	• Improved service delay • Optimized power consumption	• Not validated for complex scenarios
Hoang et al. [52]	Heuristic based method	• Completion time • Resource usage	• Improved latency rate	• Improved efficiency in service • Processing rate • High time complexity

minimising social group per execution cost under multiple constraints.

Ye et al. [59] suggested a computation offloading approach based upon a genetic algorithm. A genetic algorithm helps in allocating the computational task to fog servers while reducing overall cost and satisfying user experience in the fog computing environment.

Zhao et al. [60] suggested a task offloading method to minimise response time, energy consumption and maximise power transmission. They proposed to compute energy utilisation of different fog nodes and choose them to offload computing dynamically.

Nan et al. [61] designed an online method using Lyapunov optimization algorithm to balance computation cost and response time while offloading in fog computing. Meng et al. [62] designed a hybrid task offloading method to minimise energy uses in fog computing. They suggested dividing the loading problem into subproblems and providing a communication computation scheduling solution for each sub problem. Similarly, the heuristic based method for solving task assignment problems in fog computing is also proposed by Chamola et al. [63].

Khan et al. [64] suggested an approach for task offloading distributed mobile for computing environment to cache the content at the network's edge. It enables self-organization and caching nodes collaboratively. At the same time, Alam et al. [65] applied reinforcement learning to design offloading methods in a decentralized manner to propagate mobile code on geographical e distributed for computing. They aimed to minimise mobile users' energy consumption, execution time, and latency.

Bozorgchenani et al. [66] advocated energy aware offloading method using three components, fog nodes, cluster hat and cluster assignment for classification of fog nodes, selection of for clusters and assignment of cluster members, respectively in for computing environment.

Ahn et al. [67] introduced computational offloading and network resource allocating using two tier offloading design for optimising energy and time in the cloud computing environment.

Zhu et al. [68] suggested a task offloading approach by focusing on optimising execution time and power consumption of mobile devices. Chen et al. [69] applied game theory to obtain Nash equilibrium for proposing an effective computational offloading method. They aimed to optimise computational overhead and convergence time in a distributed way in the mobile edge computing environment. Chang et al. [70] proposed a task offloading problem using an energy efficient approach based upon queuing theory for executing network processes in the fog computing environment.

Kattepur et al. [71] used linear programming for proposing and computational offloading approach for achieving trade off between latency and energy e of mobile for the fog computing environment. The authors of [72] presented a near optimal partial offloading method based on energy consumption and processing delay of fog nodes. Xiong et al. [73] focused on the interaction between block miner and fog provider based upon Game Theory for mining tasks offloading. They propose a two staged Stackelberg game to transform resource management for computing based upon price competition in block chain consensus for maximizing profit for provider and block miners.

The above-mentioned studies of task offloading in the fog computing environment show that most researches focused on model-based methods using game theory, queueing theory, etc. Some researchers have also used heuristic based models.

Table 5 presents a comprehensive summary of task offloading based resource management strategies mentioned above.

B. Load balancing methods

In the fog computing environment, balancing the load is considered a critical issue for resource management. Load balancing methods attempt to distribute workload on fog nodes while meeting latency and energy consumption parameters. The

fog computing environment contains fog nodes with different capabilities. Accordingly, load balancing methods distribute incoming workload among fog nodes that helps to avoid overloaded and under loaded nodes while maintaining different parameters such as minimum response time.

Several methods have been proposed for load balancing in the fog computing environment. For example, Li et al. [74] proposed a self-similarity based method for balancing workload in large-scale systems of fog computing. They proposed that each fog node contains three components for monitoring internet and intranet information, a scheduler for executing distributed load balancing method and a messenger for transmitting a message across different fog nodes.

Shi et al. [75] applied Swarm optimisation methods for optimising the quality of service constraints and latency by balancing workload in a fog computing environment. Manasrah et al. [76] suggested an improvement of the service broker method for balancing work log in fog quitting environment. They utilised the differential evolution optimisation method for selecting the appropriate cloud data centre in performing the different tasks with minimum response time and cost.

He et al. [77] also applied a meta heuristic technique for balancing workload among fog nodes to minimise latency. Beraldi et al. [78] introduced a Cooperative approach for workload balancing in fog computing by reducing block in states and delaying task execution at the cloud data centre.

Ningning et al. [79] applied graph theory to develop a dynamic workload balancing approach in a fog computing environment.

Table 5. Summary of task offloading based resource management strategies

Study	Method employed	Performance metrics	Pros	Cons
Tran et al. [53]	Matching theory based method	• Running time	• Improved the running time • Improving the accuracy	• High computational complexity • Power consumption and delay not validated
Liu et al. [54]	Queuing theory, IPM-based algorithm	• Power consumption • Delay • Payment cost	• Suitable for the heterogeneous • Improved power consumption • Improved delay • Improved payment cost	• Not validated for real world scenarios • High computational complexity
Mukherjee et al. [55]	Heuristic based method	• Throughput • Carried load • Power consumption • Delay • Jitter	• Fast and cooperative offloading • Improved the delay • Improved jitter • Improved power consumption	• Not considered the fault-tolerance mechanism for fog nodes • Not suitable for outdoor applications
Wang et al. [56]	Queuing theory, and Approximate based method	• Response time	• Utilizing parked and moving • Vehicles as fog nodes • Systems • Improved response time • Supporting real-time traffic on • The internet of vehicle (iov)	• Overhead not validated • Power consumption has not validated
Xu et al. [57]	Reinforcement learning based method	• Cost	• Supports of renewable-powered systems performance • High harvesting efficiency • Improved the convergence • Speed and run-time	• Improved scalability • Lack of an appropriate • Simulation
Liu et al. [58]	Queuing theory, and Game Theory based method	• Execution cost • Power consumption, • Execution delay	• Improved social group	• Improved scalability • Overhead not validated
Ye et al. [59]	Genetic algorithm based method	• Cost • Dropping rate	• High scalability • Improved total cost	• Not validated power consumption • Not validated delay • Lack of an appropriate simulation
Zhao et al. [60]	Non-linear fractional programming based method	• Power consumption	• Considered the latency and the transmission power • Improved power consumption	• Not validated in a real-world • High computational complexity
Nan et al. [61]	Lyapunov Optimization based method	• Number • Response time, cost, • Application loss	• Improved computational complexity • Improved response time • Improved cost • Improved number of application loss	• Not validated communication power consumption
Meng et al. [62]	Closed-form based solution	• Power consumption, • Communication delay	• Improved power consumption • Considered hybrid offloading (cloud and fog) • Ignored single offloading	• Not validated in a real-world scenario • High computational complexity
Chamola et al. [63]	Heuristic based method	• Latency	• Improved of network latency • Suitable for sdn switched network	• Power consumption not validated • High computational complexity • Lack of an appropriate • Simulation
Khan et al. [64]	Game theory based method	• Cache hit rate	• Self-organize • Improving cache hits ratio	• Ignored weighting different factors

				• Not validated for content and node profile in delay-sensitive dynamic networks
Alam et al. [65]	Reinforcement Learning based method	• Response time • Power consumption	• Suitable for multi-agent systems • Improved the execution time • Improved latency	• Ignored mobility • Ignored privacy • Ignored context-aware offloading • Overhead not validated
Bozorgchenani et al. [66]	Heuristic based method	• Power consumption • Task delay • Network lifetime	• Fairness • Improved network lifetime • Improved power consumption • Improved delay	• High computational complexity • The dependency of the cluster updating frequency
Ahn et al. [67]	Heuristic based method	• Power • Wait time	• Improved power consumption • Improved latency • Improved time • Improved power expenditure	• Heterogeneity of devices ignored • High computational complexity
Zhu et al. [68]	Heuristic based method	• Power consumption, • Execution time	• Execution time • High scalability • Improved power consumption	• Not validated overhead • Ignored dynamic offloading • Ignored virtual machine migration
Chen et al. [69]	Game theory based method	• Computation overhead, • Convergence time	• High scalability • Improved the computation time	• Not considered the user mobility patterns • Not validated communication overhead
Chang et al. [70]	Queuing theory , ADMM based method	• Power consumption • Delay	• Improved power consumption • Considered heterogeneity of the queue	• Lack of an appropriate • Simulation • High computational complexity
Kattepur et al. [71]	Linear programming based method	• Power consumption • Latency	• Improved the power consumption • Improved latency • Modeling the battery discharge profiles	• Not investigated scalability • Not investigated accuracy
Bozorgchenani et al. [72]	Heuristic based method	• Power consumption • Task delay • network lifetime	• Improved network lifetime • Improved power consumption • Improved delay	• High computational complexity • The dependency of the cluster updating frequency
Xiong et al. [73]	Game theory based method	• Average optimal price • Propagation delay • Profit	• Increased the profit of fog providers • Improving the utility of miners	• Not considered competition between service providers • Not validated power consumption

They proposed abstracting the physical no graph model for reducing the number of node migrations based on graph partitioning and clustering methods.

Oueis et al. [80] propose two stage approach for workload balancing in the fog computing environment. The first phase
involves the allocation of local computing resources. The second phase enables computational clusters for different requests.

Yu et al. [81] suggested workload balancing approach info computing environment by focusing on reducing perfect hashing based upon a new data structure. Neto et al. [82] suggested a multi-tenant method for distributing load among fog nodes. They focused on particular tenants' needs, such as tenant priority and acceptance delay.

Gu et al. [83] proposed a task distribution method for medical cyber physical systems and reduced cost and improved quality of service parameters. Kapsalis et al. [84] suggested a load balancing method based upon the score function of computing resource utilisation, battery life and latency of different hosts in the fog computing environment. Xu et al. [85] reported a resource allocation method to effectively balance workload among computing nodes. Their approach consists of four components for partitioning fog service, finding extra space for computing nodes, allocating resources statically, and balancing load using global resource allocation. Verma et al. [86] used a data replication approach to develop a load-balancing framework. They attempted to minimise processing time, response and cost in the fog computing environment.

The above-mentioned studies of workload balancing in the fog computing environment show that most researchers focused on differential evolution, graph theory, linear programming, etc, for proposing load balancing approaches.

Most of the approaches are evaluated using response time and latency parameters.

Most researchers ignored energy consumption while balancing the load in the fog computing environment.

Table 6 presents a comprehensive summary of load balancing based resource management strategies mentioned above.

C. Resource allocation methods

In the fog computing environment, the resource allocation methods efficiently allocate computing resources among the graphically distributed fog nodes that are generally heterogeneous under the different quality of service requirements and other constraints. Several methods have been proposed that can be broadly categorised as auction based and optimisation based methods. The auction based methods are proposed based on market pricing methods considering demand and supply of fog nodes. In contrast, optimisation-based resource allocation methods match IoT users' cloud servers and fog nodes. They formulated the resource allocation problem as NP hard problem. They found near optimal solutions using optimisation techniques to allocate fog nodes two different IoT services with different quality of service needs.

Ni et al. [87] suggested a resource allocation method based on Petri nets in a fog computing environment by considering time and cost parameters in completing the task. Zhang et al. [88] proposed a resource allocation method in two tier heterogeneous vehicle network defined using software defined network technology and fog computing environment. Their approach provided state space function using energy conditions and allocated resources as the main field game.

Zhang et al. [89] used Stackelberg game Concept for proposing the resource allocation method. The proposed game contains many data services, data services subscribers, and limited computing resources. Do et al. [90] suggested a distributed method for resource allocating based upon the proximal algorithm. They proposed to solve the resource allocation method as a General convex optimisation method using utility and cost functions in video streaming.

Alsaffar et al. [91] applied decision tree learning based methods for efficient resource allocation in the fog computing environment. They attempted to minimise completion time, Virtual Machine capacity and service size for managing user requests in for computing environment.

Zhang et al. [92] proposed a new framework based upon the hierarchical game concept. Their framework consists of three components for operating Data Services, authorising Data Services subscribers and fog nodes. Their framework applied the Stackelberg game for interacting between data service operators and authorised data service subscribers.

Aazam et al. [93] presented a resource management technique by considering resource approximation and allocation. The targeted on estimating resources and pricing considering types of customers and devices.

Sood et al. [94] suggested a fog layer in an optical network to improve their computing abilities. They analysed using deadlock managers to design resource allocation graphs and accordingly taken required decisions in for computing environment. Naranjo et al. [95] suggested an approach for allocating resources in the fog computing environment by scheduling incoming traffic into fog nodes dynamically. Anglano et al. [96] also propose an approach using an approximation algorithm for enhancing the profit of Edge providers by handling workload fluctuations under the quality of service constraints. Jiao et al. [97] focused on auction based approach for allocating resources using an approximate algorithm in the cloud computing environment.

The above cited studies show that action-based approaches help business to business models for improving their profit, increasing fog node requests under certain case studies such as Healthcare applications. Whereas optimisation based resource allocation approaches offer an effective and cost-benefit approximation of different case studies such as time sensitive applications and real time applications. Most optimisation methods employed heuristic algorithms for developing resource allocation methods in the fog computing environment. Most researchers used iFogsim simulator for validating their approaches.

Table 7 presents a comprehensive summary of resource allocation based resource management strategies mentioned above.

D. Resource provisioning methods

Resource provisioning methods handle the workload fluctuations of a time. The workload fluctuations can result in over provisioning or under provisioning issues in the fog computing environment. Over provisioning cases result in allocating additional resources than the actual requirement. Whereas under provisioning cases consist of allocating less resources than the actual requirement of IoT users in the computing environment. Under provision scenarios can cors service level agreement violations and hence loss of IoT users.

Therefore, it is required to use resource provisioning methods for dynamically provision computing resources as per the requirement of IoT users while maintaining minimum cost and meeting quality of service requirements as per service level agreement.

Several approaches have been proposed for effective resource provisioning to meet fluctuating demands of IoT users. For example, El Kafhali et al. [98] proposed a resource provisioning method based upon queuing theory concepts that help to find an appropriate quantity of fog resources for different IoT devices. Their model consists of edge nodes, cloud gateways, and cloud data centre. Wang et al. [99] designed a Framework for deploying workload that dynamically adds and removes fog computing resources to address the issue of fluctuating workload in edge nodes of the computing environment. Tseng et al. [100] suggested a resource provisioning method using lightweight fuzzy logic concepts for industrial applications in fog computing.

Table 6. Summary of load balancing based resource management strategies

Study	Method employed	Performance metrics	Pros	Cons
Li et al. [74]	Threshold based method	• Resource usage, • Execution time	• Thresholds dynamically • High scalability • Tunning proper load • Improved overhead	• Bottleneck ignored • Power consumption not validated • Throughput not validated
Shi et al. [75]	MPSO based method	• Latency	• Improved service latency • Suitable for real-time mobile • Applications	• Improved scalability not validated in real world scenario
Manasrah et al. [76]	Differential Evolution based method	• Response time • Cost	• Improved response time • Improved cost • Improved latency	• Lack of data privacy mechanism • Not considered user application priorities
He et al. [77]	Graph partitioning Theory based method	• Number of move nodes • Run time	• Improved service latency • Improved computational complexity	• Not validated in a real world scenarios • Power usage not validated
Beraldi et al. [78]	fireworks algorithm (FWA) based method	• Latency	• Improved service delay • Improved blocking probability	• Power consumption not • Validated
Ningning et al. [79]	Parallel Othello Group (FWA)	• Throughput	• Improved the migration overhead • Easy to implement	• Lack of different weighting simulation factors • Ignored characteristics of the graph repartitioning
Oueis et al. [80]	Heuristic based method	• Tenant maximum • Acceptable delay • Tenant priority	• Improved computational complexity • Customizable to specific applications and network requirements	• Bottleneck • Scalability
Yu et al. [81]	Heuristic based method	• Latency • Power consumption • User satisfaction ratio	• High scalability • Memory-efficient	• Power consumption not validated • Bottleneck
Neto et al. [82]	Heuristic based method	• Throughput • Turnaround time	• Improved delay and priority	• Overhead investigated • Ignored of the disk i/o operations
Gu et al. [83]	Linear Programming (FWA)	• Total cost	• Optimized the total cost • Improved computational • Complexity	• Overhead not investigated
Kapsalis et al. [84]	Heuristic based method	• Execution time delay • Number of tasks	• Improved the execution time • Improved delay • Improved throughput • Improved overhead communication protocol	• Not validated for a real case
Xu et al. [85]	Heuristic based method	• Load-balance variance • Resource usage	• Improved the resource • Computing node priority on the load-balance variance for each usage, and throughput	• Service migration not validated • Power consumption not validated • delay not validated
Verma et al. [86]	Heuristic based method	• Response time • Cost	• Improved time • Improved cost	• Lack of an appropriate simulation • Ignored security issues

Dos Santos et al. [101] used integer linear programming to optimise many e conflicting objectives bile provisioning computing resources in the computing environment. They considered latency, energy consumption and service migration as conflicting objectives in IoT applications in their proposed approach. Arkian et al. [102] suggested a cost effective method for computing resource provisioning in IoT applications in Association with task distribution and placement of virtual machines. They focus on provisioning optimising resource provisioning and quality of service requirements.

Vinueza Naranjo et al. [103] also developed a Framework for managing computing resources dynamically based upon bin packing heuristic method. They attempted to minimise energy utilisation while maintaining the quality of service requirements.

Ostberg et al. [104] presented a framework consisting of four mud used for a reliable capacity provisioning of computing resources in the fog computing environment. The significant components are collector, application modeller, work modeller and the optimizer for distributing fog applications among different fog nodes.

Zanni et al. [105] proposed a geometric monitoring system based on dynamic scaling. They validated their model based

upon a geographically distributed system and demonstrated their solution to be latency effective. Skarlat et al.

[106] also designed a Framework for resource provisioning consisting of four modules fog cells, fog colonies, fog-cloud controller middleware.

Russo Russo et al. [107] suggested a hierarchical framework for managing elastic data stream processing applications. Their framework consists of two components for managing deployment operations and fog computing resources.

It can be concluded from the above mentioned studies of resource provisioning methods that most researchers used heuristic based algorithms to develop resource provisioning methods while meeting the quality of service requirements. They mainly focused on latency, cost and delay parameters in their studies. Some researchers have also focused on energy consumption and processor utilisation.

Table 8 presents a comprehensive summary of resource provisioning based resource management strategies mentioned above.

5. Resource Management Issues

It can be e seen from above mentioned sections that many efforts have been invested in managing computing resources effectively in the fog computing environment. However, many issues require immediate addressing in different aspects. Different researches focused on different aspects such as application placement, computing resource provisioning, computing resource scheduling and task offloading. The

Table 7. Summary of resource allocation based resource management strategies

Study	Method employed	Performance metrics	Pros	Cons
Ni et al. [87]	Priced Timed Petri nets (FWA)	• Cost, • Makespan • Number of completing task	• Improved cost • Improved makespan	• Not analyzed time • Not analyzed omitting fairness • Not analyzed correctness evaluation
Zhang et al. [88]	Game theory based method	• Utility of fn • Dss • Dso	• High utility • Improved delay	• Not considered evaluating time • Not considered analyzing cost
Zhang et al. [89]	Game theory based method	• Time • Power • Interference factor	• Optimized cost	• Not evaluated scalability • Not considered analyzing time
Do et al. [90]	ADMM based method	• Convergence • Speed • Carbon footprint	• Improved time	• Not compared with other algorithms
Alsaffar et al. [91]	Heuristic based method	• Processing time • Number of vm	• Optimized the number of vm • Improved response time	• High cost • Improved scalability
Zhang et al. [92]	Game theory based method	• Utility of adss • Dso	• Improved delay	• Not considered evaluating time • Not considered analyzing cost

Aazam et al. [93]	Heuristic based method	• Service price, resource • Estimation	• Improved price	• Not considered evaluating time • Not considered analyzing cost
Sood et al. [94]	Social Network Analysis (SNA) And Rule based method	• Power consumption • Latency • Number of deadlocks detected • Resource usage	• Improved latency • Optimized the number of vms	• Not considered bandwidth • Not considered time
Naranjo et al. [95]	Heuristic based method	• Power consumption, • Turned on servers	• Improved delay	• Not considered evaluating time • Not considered analyzing cost
Anglano et al. [96]	Approximation algorithm based method	• Profit	• Improved delay • Maximization profit	• Not considered workload prediction • Lack of appropriate simulations
Jiao et al. [97]	Approximate algorithm based method	• Social welfare	• Improved delay	• High time complexity

significant issues in resource management in fog computing environments are described below.

The authors of [1] described that application placement in fog computing environment plays a significant role in managing resources efficiently. They described that application placement could be categorised into three classes, centralised, decentralized, and hierarchical classes. In centralised application placement methods, there is a requirement of a centralised broker that requires information from all devices in the fog computing environment, such as fog devices, cloud, crimes and IoT services. A centralised broker takes global optimisation decisions based upon the received information. In decentralized application placement, decentralized brokers have partial information and are suitable for a small number of fog computing devices. Centralised application placement methods suffer from the limitation of exhibition overhead and fault tolerance due to global knowledge transmission from all fog computing environment devices to centralised brokers for taking globally optimised solutions. It also suffers from single point failure due to centralise diseases of the centralised broker. Transmission of information from all fog computing devices increases the network traffic with the increase in the number of IoT devices.

Resource scheduling methods happen categorised into three main classes, static, dynamic and hybrid. Static resource scheduling methods involves scheduling decisions before the arrival of different tasks. It indicates that static resource scheduling methods contain information about computational requirements and available computing resources in advance. However, this may not be the scenario in many heterogeneous systems, particularly the computing environment. Static resource scheduling methods cannot guarantee optimal scheduling of resources in the fog computing environment.

In the case of the dynamic scheduling methods, computing resources are allocated after submitting tasks as per their requirements. Dynamic scheduling methods do not assume any prior information about the arrival of tasks. Whereas hybrid scheduling methods mix both Static and dynamic scheduling methods in scheduling different tasks in for computing environments as per their requirements.Task offloading method

Load balancing methods can be divided into three major categories as centralised, decentralized and hybrid. Centralised load balancing methods use a central controller as a load balancer that requires global knowledge of power resources and IoT user requirements. These approaches suffer from a single point of failure and are not scalable in nature.

Decentralized load balancing methods attach the limitation of a single point of failure in workload balancing. At the same time, hybrid methods use both centralised and decentralized approaches in balancing workload of the fog computing environment.

Resource provisioning methods have been classified as per their strategies used, reactive, Proactive and hybrid methods. Reactive methods do not involve any kind of prediction. These methods respond as per the current status of the fog computing system. Proactive resource provisioning methods involve prediction methods for approximating future demands based upon historical data of IoT applications and provision computing resources accordingly to minimise overloaded and under loaded nodes in the fog computing environment. Hybrid resource provisioning methods benefit both reactive and proactive to handle workload fluctuations in the fog computing environment.

Table 8. Summary of resource provisioning based resource management strategies

Study	Method employed	Performance metrics	Pros	Cons
El Kafhali et al. [98]	Queuing theory based method	• Response messages time • Throughput • Drop rate • Number of cpu usage	• Improved throughput • Improved cpu usage • Improved the time • Improved loss rate • Used an analytical queuing model	• Distribution overhead not investigated • The dependency of the request arrival rate
Wang et al. [99]	Heuristic based method	• Frequency • Latency, data traffic, • Communication	• Improved overhead • Improved the service latency • Easy to implementation	• Not considered dynamic priorities of applications • Not considered migrating applications between nodes
Tseng et al. [100]	Fuzzy theory based method	• Delay • Error rate • Total operating expenses	• High accuracy • Improved overhead • Improved cost	• Not considered live migrations • Not considered power consumption
Dos Santos et al. [101]	Linear Programming based method	• Ratio of service migration • Ratio of active computation gateway • Hop count • Ratio of active • Computation nodes	• Improved computational complexity • Improved migration service • Improved latency • Improved power consumption	• Not investigated overhead
Arkian et al. [102]	Linear Programming based method	• Power consumption • Service latency • Cost	• Optimized the overall cost • Improved power consumption • Supports IoT crowd-sensing applications	• Not validated in a real-world scenario • Not considered privacy-preserving data • Analytics capabilities
Vinueza Naranjo et al. [103]	Heuristic based method	• Power consumption	• Improved computational complexity • Supports container-based virtualization	• Latency not validated • Workload not considered
Zanni et al. [105]	Heuristic based method	• Latency	• Suitable for mobile applications • High scalability • Supports container-based virtualization	• Lack of an appropriate • Simulation • Not considered resource • Provisioning algorithms
Skarlat et al. [106]	Heuristic based method	• Time • Delay • Makespan • Cost	• Improved the service latency • Improved cost	• Not considered the fault-tolerance mechanism • Lack of an appropriate simulation
Russo Russo et al. [107]	Reinforcement Learning based method	• Response time • Cost • Number of active nodes	• Avoids resource wastage • Supports the application-level elasticity	• Not considered the resource estimation mechanism

6. Conclusion

This paper aims to present Resource Management strategies in the fog computing environment. It highlights different methods and frameworks used for managing computing resources and presents their issues in the fog computing environment. Fog computing technology help to develop an effective solution for IoT based applications that are delay sensitive. It enables the processing of extensive data near the IoT devices, resulting in reduction of power consumption, latency, network traffic etc, compared to the processing at the cloud server.

However, due to the limited processing capability of fog nodes, it becomes necessary to develop efficient resource management strategies in considering different dimensions such as application placement, computing resource provisioning, computing resource scheduling, task offloading while meeting the quality of service requirement fog computing environment.

To that end, this paper presented resource management strategies in the computing environment and highlighted different issues requiring immediate addressing of the research community in the field. It compared different resource management strategies in different dimensions to access the current research status in resource management in the fog computing environment.

REFERENCES

[1] Ghobaei-Arani, M., Souri, A., & Rahmanian, A. A. (2020). Resource management approaches in fog computing: a comprehensive review. Journal of Grid Computing, 18 (1), 1-42.

[2] Ghobaei-Arani, M., Shamsi, M., Rahmanian, A.A.: An efficient approach for improving virtual machine placement in cloud computing environment. J. Exp. Theor. Artif. Intell. 29 (6), 1149 – 1171 (2017)

[3] Souri, A., Asghari, P., Rezaei, R.: Software as a service based CRM providers in the cloud computing: challenges and technical issues. J. Serv. Sci. Res. 9 (2), 219 – 237 (2017)

[4] Ghobaei-Arani, M., Rahmanian, A.A., Aslanpour, M.S., Dashti, S.E.: CSA-WSC: cuckoo search algorithm for web service composition in cloud environments. Soft. Comput. 22 (24), 8353 – 8378 (2018)

[5] Ghobaei-Arani, M., Rahmanian, A.A., Shamsi, M., Rasouli-Kenari, A.: A learning-based approach for virtual machine placement in cloud data centers. Int. J. Commun. Syst. 31 (8), e3537 (2018)

[6] Manasrah, A.M., Gupta, B.: An optimized service broker routing policy based on differential evolution algorithm in fog/cloud environment. Clust. Comput. 22 (Supplement 1), 1639 – 1653 (2017)

[7] Mouradian, C., et al.: A comprehensive survey on fog computing: state-of-the-art and research challenges. IEEE Commun. Surv. Tutorials. (2017)

[8] Hong, C.-H. and B. Varghese, Resource Management in Fog/Edge Computing: A Survey. arXiv preprint arXiv: 1810.00305, (2018)

[9] Aazam, M., Zeadally, S., Harras, K.A.: Offloading in fog computing for IoT: review, enabling technologies, and research opportunities. Futur. Gener. Comput. Syst. 87, 278 – 289 (2018)

[10] Masip-Bruin, X., Marin-Tordera, E., Jukan, A., Ren, G.J.: Managing resources continuity from the edge to the cloud: architecture and performance. Futur. Gener. Comput. Syst. 79, 777 – 785 (2018)

[11] Toczé, K., Nadjm-Tehrani, S.: A taxonomy for management and optimization of multiple resources in edge computing. Wirel. Commun. Mob. Comput. 2018, 1 – 23 (2018)

[12] Dias de Assunção, M., da Silva Veith, A., Buyya, R.: Distributed data stream processing and edge computing: A survey on resource elasticity and future directions. J. Netw. Comput. Appl. 103, 1 – 17 (2018)

[13] Hong, C.-H. and B. Varghese, Resource Management in Fog/Edge Computing: A Survey. arXiv preprint arXiv: 1810.00305, (2018)

[14] Naha, R.K.; Garg, S.; Georgakopoulos, D.; Jayaraman, P.P.; Gao, L.; Xiang, Y.; Ranjan, R. Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions. IEEE Access 2018, 6, 47980 – 48009

[15] Bendechache, M.; Svorobej, S.; Takako Endo, P.; Lynn, T. Simulating Resource Management across the Cloud-to-Thing Continuum: A Survey and Future Directions. Future Internet 2020, 12, 95

[16] Aslanpour, M.S.; Gill, S.S.; Toosi, A.N. Performance evaluation metrics for cloud, fog, and edge computing: A review, taxonomy, benchmarks and standards for future research. Internet Things 2020, 12, 100273

[17] Salaht, F.A.; Desprez, F.; Lebre, A. An Overview of Service Placement Problem in Fog and Edge Computing. ACM Comput. Surv. 2020, 53, 1 – 35

[18] Agarwal, S.; Yadav, S.; Yadav, A.K. An efficient architecture and algorithm for resource provisioning in fog computing. Int. J. Inf. Eng. Electronic Bus. (IJIEEB) 2016, 8, 48 – 61

[19] Souri, A., Norouzi, M.: A state-of-the-art survey on formal verification of the internet of things applications. J. Serv. Sci. Res. 11 (1), 47 – 67 (2019)

[20] Ghobaei-Arani, M., et al.: A moth-flame optimization algorithm for web service composition in cloud computing: simulation and verification. Software: Practice and Experience (SPE). 48 (10), 1865 – 1892 (2018)

[21] Dastjerdi, A.V., et al., Fog computing: Principles, architectures, and applications, in Internet of Things. Elsevier. p. 61 – 75 (2016)

[22] Mijuskovic, A., Chiumento, A., Bemthuis, R., Aldea, A., & Havinga, P. (2021). Resource management techniques for cloud/fog and edge computing: An evaluation framework and classification. Sensors, 21 (5), 1832.

[23] Javaid, S.; Javaid, N.; Saba, T.; Wadud, Z.; Rehman, A.; Haseeb, A. Intelligent resource allocation in residential buildings using consumer to fog to cloud based framework. Energies 2019, 12, 815

[24] Xu, X.; Fu, S.; Cai, Q.; Tian, W.; Liu, W.; Dou, W.; Sun, X.; Liu, A.X. Dynamic resource allocation for load balancing in fog environment. Wirel. Commun. Mob. Comput. 2018, 2018

[25] Taneja, M.; Davy, A. Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm. In Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 8 – 12 May 2017; pp. 1222 – 1228

[26] Skarlat, O., Nardelli, M., Schulte, S., Borkowski, M., Leitner, P.: Optimized IoT service placement in the fog. SOCA. 11 (4), 427 – 443 (2017)

[27] Venticinque, S., Amato, A.: A methodology for deployment of IoT application in fog . J . Ambient. Intell. Humaniz. Comput. 10 (5), 1955 – 1976 (2019)

[28] Mahmoud, M.M.E., Rodrigues, J.J.P.C., Saleem, K., al Muhtadi, J., Kumar, N., Korotaev, V.: Towards energy aware fog-enabled cloud of things for healthcare. Comput. Electr. Eng. 67, 58 – 69 (2018)

[29] Yangui, S., et al. A platform as-a-service for hybrid cloud / fog environments. In Local and Metropolitan Area Networks (LANMAN), 2016 IEEE International Symposium on. IEEE (2016)

[30] Yigitoglu, E., et al. Foggy: A Framework for Continuous Automated IoT Application Deployment in Fog Computing. In AI & Mobile Services (AIMS), 2017 IEEE International Conference on. IEEE (2017)

[31] Minh, Q.T., et al. Toward service placement on fog computing landscape. In Information and Computer Science, 2017 4th NAFOSTED Conference on. IEEE (2017)

[32] Saurez, E., et al., Incremental deployment and migration of geo distributed situation awareness applications in the fog. p. 258 – 269 (2016)

[33] Brogi, A., Forti, S.: QoS-aware deployment of IoT applications through the fog. IEEE Internet Things J. 4 (5), 1185 – 1192 (2017)

[34] Yao, H., Bai, C., Xiong, M., Zeng, D., Fu, Z.: Heterogeneous cloudlet deployment and user-cloudlet association toward cost effective fog computing. Concurrency and Computation: Practice and Experience (CCPE). 29 (16), e3975 (2017)

[35] Yousefpour, A., et al., QoS-aware Dynamic Fog Service Provisioning. arXiv preprint arXiv:1802.00800 (2018)

[36] Mahmud, R., Ramamohanarao, K., Buyya, R.: Latency aware application module Management for fog Computing Environments. ACM Trans. Internet Technol. 19 (1), 1 – 21 (2018)

[37] Naranjo, P.G.V., et al., FOCAN: A Fog-supported Smart City Network Architecture for Management of Applications in the Internet of Everything Environments. arXiv preprint arXiv:1710.01801, (2017)

[38] Mahmud, R., Srirama, S.N., Ramamohanarao, K., Buyya, R.: Quality of experience (QoE)-aware placement of applications in fog computing environments. J. Parallel Distrib. Comput. 132, 190 – 203 (2019)

[39] Velasquez, K., et al.: Service placement for latency reduction in the internet of things. Ann. Telecommun. 72 (1 – 2), 105 – 115 (2016)

[40] Selimi, M., Cerdà-Alabern, L., Freitag, F., Veiga, L., Sathiaseelan, A., Crowcroft, J.: A lightweight service placement approach for

community network micro-clouds. Journal of Grid Computing (GRID). 17 (1), 169 – 189 (2019)

[41] Bitam, S., Zeadally, S., Mellouk, A.: Fog computing job scheduling optimization based on bees swarm. Enterprise Information Systems (EIS). 12 (4), 373 – 397 (2017)

[42] Sun, Y., Lin, F., Xu, H.: Multi-objective optimization of resource scheduling in fog computing using an improved NSGA-II. Wirel. Pers. Commun. 102 (2), 1369 – 1385 (2018)

[43] De Benedetti, M., et al.: JarvSis: a distributed scheduler for IoT applications. Clust. Comput. 20 (2), 1775 – 1790 (2017)

[44] Cardellini, V., et al. On QoS-aware scheduling of data stream applications over fog computing infrastructures. In Computers and Communication (ISCC), 2015 IEEE Symposium on. IEEE (2015)

[45] Rahbari, D. and M. Nickray. Scheduling of Fog Networks with Optimized Knapsack by Symbiotic Organisms Search. In 2017 21st Conference of Open Innovations Association (FRUCT). Finland: IEEE (2017)

[46] Zeng, D., Gu, L., Guo, S., Cheng, Z., Yu, S.: Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system. IEEE Trans. Comput. 65 (12), 3702 – 3712 (2016)

[47] Pham, X.-Q. and E.-N. Huh. Towards task scheduling in a cloud-fog computing system. In Network Operations and Management Symposium (APNOMS), 2016 18th Asia-Pacific. IEEE (2016)

[48] Fan, J., et al. Deadline-Aware Task Scheduling in a Tiered IoT Infrastructure. in GLOBECOM 2017 – 2017 IEEE Global Communications Conference. Singapore: IEEE (2017)

[49] Sun, Y., Zhang, N.: A resource-sharing model based on a repeated game in fog computing. Saudi journal of biolog ical sciences (SJBS). 24 (3), 687 – 694 (2017)

[50] Chen, X., Wang, L.: Exploring fog computing-based adaptive vehicular data scheduling policies through a compositional formal method—PEPA. IEEE Commun. Lett. 21 (4), 745 – 748 (2017)

[51] Deng, R., et al.: Optimal workload allocation in fog-cloud computing towards balanced delay and power consumption. IEEE Internet Things J. 3 (6), 1171 – 1181 (2016)

[52] Hoang, D. and T.D. Dang, FBRC: Optimization of task Scheduling in Fog-Based Region and Cloud. 2017: p. 1109 – 1114

[53] Tran, D.H., Tran, N.H., Pham, C., Kazmi, S.M.A., Huh, E.N., Hong, C.S.: OaaS: offload as a service in fog net- works. Computing. 99 (11), 1081 – 1104 (2017)

[54] Liu, L., Chang, Z., Guo, X., Mao, S., Ristaniemi, T.: Multiobjective optimization for computation offloading in fog computing. IEEE Internet Things J. 5 (1), 283 – 294 (2018)

[55] Mukherjee, A., Deb, P., de, D., Buyya, R.: C2OF2N: a low power cooperative code offloading method for femtolet- based fog network. J. Super comput. 74 (6), 2412 – 2448 (2018)

[56] Wang, X., Ning, Z., Wang, L.: Offloading in internet of vehicles: a fog-enabled real-time traffic management sys- tem. IEEE Trans. Ind. Inf. 14 (10), 4568 – 4578 (2018)

[57] Xu, J. and S. Ren. Online learning for offloading and auto scaling in renewable- powered mobile edge computing. In Global Communications Conference (GLOBECOM), 2016 IEEE. IEEE (2016)

[58] Liu, L., Z. Chang, and X. Guo, Socially-aware Dynamic Computation Offloading Scheme for Fog Computing System with Energy Harvesting Devices. IEEE Internet Things J.. p. 1 – 1 (2018)

[59] Ye, D., et al., Scalable Fog Computing with Service Offloading in Bus Networks. p. 247 – 251 (2016)

[60] Zhao, X., L. Zhao, and K. Liang. An Energy Consumption Oriented Offloading Algorithm for Fog Computing. In International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness. Springer (2016)

[61] Nan, Y., Li, W., Bao, W., Delicato, F.C., Pires, P.F., Zomaya, A.Y.: A dynamic tradeoff data processing frame- work for delay-sensitive applications in cloud of things systems. J. Parallel Distrib. Comput. 112, 53 – 66 (2018)

[62] Meng, X., Wang, W., Zhang, Z.: Delay-constrained hybrid computation offloading with cloud and fog computing. IEEE (ACCESS). 5, 21355 – 21367 (2017)

[63] Chamola, V., C.-K. Tham, and G.S. Chalapathi. Latency aware mobile task assignment and load balancing for edge cloudlets. In Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on. IEEE (2017)

[64] Khan, J.A., C. Westphal, and Y. Ghamri-Doudane. Offloading Content with Self-organizing Mobile Fogs. In Teletraffic Congress (ITC 29), 2017 29th International. IEEE (2017)

[65] Alam, M.G.R., Y.K. Tun, and C.S. Hong. Multi-agent and reinforcement learning based code offloading in mobile fog. In Information Networking (ICOIN), 2016 International Conference on. IEEE (2016)

[66] Bozorgchenani, A., D. Tarchi, and G.E. Corazza. An Energy-Aware Offloading Clustering Approach (EAOCA) in fog computing. In Wireless Communication Systems (ISWCS), 2017 International Symposium on. IEEE (2017)

[67] Ahn,S.,M. Gorlatova, and M. Chiang. Leveraging fog and cloud computing for efficient computational offloading. In Undergraduate Research Technology Conference (URTC), 2017 IEEE MIT. IEEE (2017)

[68] Zhu, Q., Si, B., Yang, F., Ma, Y.: Task offloading decision in fog computing system. China Communications (Chinacom). 14 (11), 59 – 68 (2017)

[69] Chen, X., Jiao, L., Li, W., Fu, X.: Efficient multi-user computation offloading for Mobile-edge cloud computing. IEEE/ACM Trans. Networking. 24 (5), 2795 – 2808 (2016)

[70] Chang, Z., et al. Energy Efficient Optimization for Computation Offloading in Fog Computing System. In GLOBECOM 2017-2017 IEEE Global Communications Conference. IEEE (2017)

[71] Kattepur, A., et al. Resource constrained offloading in fog computing. In Proceedings of the 1st Workshop on Middleware for Edge Clouds & Cloudlets. ACM (2016)

[72] Bozorgchenani, A., D. Tarchi, and G.E. Corazza. An Energy and Delay-Efficient Partial Offloading Technique for Fog Computing Architectures. In GLOBECOM 2017- 2017 IEEE Global Communications Conference. IEEE (2017)

[73] Xiong, Z., et al.: Cloud/fog computing resource manage- ment and pricing for blockchain networks. IEEE Internet Things J. 6 (3), 4585 – 4600 (2018)

[74] Li, C., Zhuang, H., Wang, Q., Zhou, X.: SSLB: self- similarity- based load balancing for large- scale fog computing. Arab. J. Sci. Eng. 43 (12), 7487 – 7498 (2018)

[75] Shi, C., Z. Ren, and X. He, Research on Load Balancing for Software Defined Cloud-Fog Network in Real-Time Mobile Face Recognition. 210: p. 121 – 131 (2018)

[76] Manasrah, A.M., A.a. Aldomi, and B.B. Gupta, An opti- mized service broker routing policy based on differential evolution algorithm in fog/cloud environment. Cluster Computing, (2017)

[77] He, X., Ren, Z., Shi, C., Fang, J.: A novel load balancing strategy of software-defined cloud/fog networking in the internet of vehicles. China Communications (Chinacom). 13 (2), 140 – 149 (2016)

[78] Beraldi, R., A. Mtibaa, and H. Alnuweiri. Cooperative load balancing scheme for edge computing resources. In Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on. IEEE (2017)

[79] Ningning, S., Chao, G., Xingshuo, A., Qiang, Z.: Fog computing dynamic load balancing mechanism based on graph repartitioning. China Communications (Chinacom). 13 (3), 156 – 164 (2016)

[80] Oueis, J., E.C. Strinati, and S. Barbarossa. The fog balancing: Load distribution for small cell cloud comput- ing. In Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st. IEEE (2015)

[81] Yu,Y., X. Li, and C. Qian. SDLB: A Scalable and Dynamic Software Load Balancer for Fog and Mobile Edge Computing. In Proceedings of the Workshop on Mobile Edge Communications. ACM (2017)

[82] Neto, E.C.P., G. Callou, and F. Aires. An algorithm to optimise the load distribution of fog environments. In Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on. . IEEE (2017)

[83] Gu, L., Zeng, D., Guo, S., Barnawi, A., Xiang, Y.: Cost efficient resource management in fog computing supported medical cyber-physical system. IEEE Trans. Emerg. Top. Comput. 5 (1), 108 – 119 (2017)

[84] Kapsalis, A., Kasnesis, P., Venieris, I.S., Kaklamani, D.I., Patrikakis, C.Z.: A cooperative fog approach for effective workload balancing. IEEE Cloud Computing. 4 (2), 36 – 45 (2017)

[85] Xu, X., Fu, S., Cai, Q., Tian, W., Liu, W., Dou, W., Sun, X., Liu, A.X.: Dynamic resource allocation for load balancing in fog environment. Wirel. Commun. Mob. Comput. 2018, 1 – 15 (2018)

[86] Verma, S., et al. An efficient data replication and load balancing technique for fog computing environment. In Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on. IEEE (2016)

[87] Ni, L., Zhang, J., Jiang, C., Yan, C., Yu, K.: Resource allocation strategy in fog computing based on priced timed petri nets. IEEE Internet Things J. 4 (5), 1216 – 1228 (2017)

[88] Zhang, Y., et al., Resource Allocation in Software Defined Fog Vehicular Networks. 2017: p. 71 – 76

[89] Zhang, H., Xiao, Y., Bu, S., Niyato, D., Yu, F.R., Han, Z.: Computing resource allocation in three-tier IoT fog net- works: a joint optimization approach combining Stackelberg game and matching. IEEE Internet Things J. 4 (5), 1204 – 1215 (2017)

[90] Do, C.T., et al. A proximal algorithm for joint re- source allocation and minimizing carbon footprint in geo-distributed fog computing. In Information Networking (ICOIN), 2015 International Conference on. IEEE (2015)

[91] Alsaffar,A.A.,Pham,H.P.,Hong,C.S.,Huh,E.N.,Aazam, M.: An architecture of IoTservice delegation and resource allocation based on collaboration between fog and cloud computing. Mob. Inf. Syst. 2016, 1 – 15 (2016)

[92] Zhang, H., Zhang, Y., Gu, Y., Niyato, D., Han, Z.: A hierarchical game framework for resource management in fog computing. IEEE Commun. Mag. 55 (8), 52 – 57 (2017)

[93] Aazam, M., et al., IoT resource estimation challenges and modeling in fog, in Fog Computing in the Internet of Things, Springer. p. 17 – 31 (2018)

[94] Sood, S.K., Singh, K.D.: SNA based resource optimization in optical network using fog and cloud computing. Opt. Switch. Netw. 33 (July), 114 – 121 (2017)

[95] Naranjo, P.G., et al.: Fog over virtualized IoT: new oppor- tunity for context-aware networked applications and a case study. Appl. Sci. 7 (12), 1325 (2017)

[96] Anglano, C.,M. Canonico, and M. Guazzone. Profit-aware resource management for edge computing systems. In Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking. ACM (2018)

[97] Jiao,Y.,etal.:Auction mechanisms in cloud / fog computing resource allocation for public Block chain networks. IEEE Trans. Parallel Distrib. Syst. 30 (9), 1975 – 1989 (2018)

[98] El Kafhali, S., Salah, K.: Efficient and dynamic scaling of fog nodes for IoT devices. J. Super comput. 73 (12), 5261 – 5284 (2017)

[99] Wang, N., et al., ENORM: A Framework For Edge Node Resource Management. IEEE Transactions on Services Computing. Early access: p. 1 – 1 (2017)

[100] Tseng, F.-H., Tsai, M.S., Tseng, C.W., Yang, Y.T., Liu, C.C., Chou, L.D.: A lightweight auto-scaling mechanism for fog computing in industrial applications. IEEE Trans. Ind. Inf. 14 (10), 4529 – 4537 (2018

[101] Dos Santos, X., et al. Resource provisioning for IoT appli- cation services in Smart Cities. in CNSM2017, the 13e International Conference on Network and Service Management. (2017)

[102] Arkian, H.R., Diyanat, A., Pourkhalili, A.: MIST: fog- based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications. J. Netw. Comput. Appl. 82, 152 – 165 (2017)

[103] Vinueza Naranjo, P.G., E. Baccarelli, and M. Scarpiniti, Design and energy-efficient resource management of virtualized networked Fog architectures for the real-time support of IoT applications. J. Super comput., 2018. 74 (6): p. 2470 – 2507

[104] Östberg, P.-O., et al. Reliable capacity provisioning for distributed cloud/edge/fog computing applications. In Networks and Communications (EuCNC), 2017 European Conference on. IEEE (2017)

[105] Zanni, A., et al. Elastic Provisioning of Internet of Things Services Using Fog Computing: An Experience Report. In 2018 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud). IEEE (2018)

[106] Skarlat, O., et al. Resource provisioning for IoTservices in the fog. In Service-Oriented Computing and Applications (SOCA), 2016 IEEE 9th International Conference on. IEEE (2016)

[107] Russo Russo, G., Nardelli, M., Cardellini, V., Lo Presti, F.: Multi-level elasticity for wide-area data streaming systems: a reinforcement learning approach. Algorithms. 11 (9), 134 (2018)