applied sciences

MDPI

Article

An Optimal Task Assignment Strategy in Cloud-Fog Computing Environment

Jung-Fa Tsai [1], Chun-Hua Huang [1] and Ming-Hua Lin [2,*]

[1] Department of Business Management, National Taipei University of Technology, Taipei 10608, Taiwan; jftsai@ntut.edu.tw (J.-F.T.); James-huang@cm-groups.com (C.-H.H.)
[2] Department of Urban Industrial Management and Marketing, University of Taipei, Taipei 11153, Taiwan
* Correspondence: mhlin@utaipei.edu.tw

Abstract: With the advent of the Internet of Things era, more and more emerging applications need to provide real-time interactive services. Although cloud computing has many advantages, the massive expansion of the Internet of Things devices and the explosive growth of data may induce network congestion and add network latency. Cloud-fog computing processes some data locally on edge devices to reduce the network delay. This paper investigates the optimal task assignment strategy by considering the execution time and operating costs in a cloud-fog computing environment. Linear transformation techniques are used to solve the nonlinear mathematical programming model of the task assignment problem in cloud-fog computing systems. The proposed method can determine the globally optimal solution for the task assignment problem based on the requirements of the tasks, the processing speed of nodes, and the resource usage cost of nodes in cloud-fog computing systems.

Keywords: task assignment strategy; cloud-fog computing; mathematical programming model; linear transformation technique

Citation: Tsai, J.-F.; Huang, C.-H.; Lin, M.-H. An Optimal Task Assignment Strategy in Cloud-Fog Computing Environment. *Appl. Sci.* **2021**, *11*, 1909. https://doi.org/10.3390/app11041909

Academic Editor: Paula Fraga-Lamas

Received: 30 January 2021
Accepted: 18 February 2021
Published: 22 February 2021

1. Introduction

In the era of the Internet of Things (IoT), many emerging applications need to provide real-time responses and interactions. According to the report by market insights firm IoT Analytics, global IoT device connections are estimated to surpass non-IoT device connections and reach 11.7 billion in 2020 [1]. The International Data Corporation (IDC) predicts that 75% of 55.7 billion devices worldwide will be connected to an IoT platform by 2025. IDC also estimates that the data generated from the IoT devices will grow from 18.3 ZB in 2019 to 73.1 ZB by 2025 [2].

In various industries, such as transportation, oil and gas, manufacturing, mining, and utilities, a short response time plays a vital role in improving the output, boosting service levels, and increasing the safety. The data sensed from the IoT devices often requires a rapid and real-time analysis of the data. Consequently, an appropriate infrastructure must be designed to deal with the rapid growth of the IoT data for making a timely and correct decision during event detection [3]. Cisco [3] listed the main requirements for the processing of the IoT data; a well-designed task assignment strategy can satisfy the following requirements:

- latency decrement
- network bandwidth conservation
- data movement to the best place for processing

Although cloud computing has many advantages, the massive expansion of the IoT devices and the explosive growth of data may induce network congestion and add network latency. Conventional cloud computing architectures that transmit all data from the network edge to the central data center for processing cannot meet all of the above requirements. Transmitting the huge amount of data from the IoT devices to the cloud imposes a significantly heavy burden on network performance. This situation also results in unreliable network latency or uncertain response time for end-users [4,5].

Edge computing has a decentralized architecture that assigns processing tasks to the edge in the network to reduce the network delay. Fog computing transforms network edge devices into parts of a distributed computing architecture to implement IoT applications such as medical and healthcare, building and home automation, traffic control, environmental monitoring, energy management, transportation networks, etc. [3,4]. Compared to pure cloud computing, edge computing and fog computing perform better within the aspects of data transmission speed, privacy and security, limited bandwidths, and data control [6]. Due to the advancements in information technology, conventional network edge devices—for instance, routers, gateways, workstations, and personal computers—have become increasingly powerful in the context of the processing capability, storage space, and communication capability. The resources not only can be utilized by their owners but help to push data handling to the network edge [5].

This paper focuses on the optimal task assignment strategy that minimizes the execution time and operating costs in a cloud-fog computing environment. Nguyen et al. [4] constructed a nonlinear mathematical programming model to treat the task assignment problem in a cloud-fog computing environment for the IoT. The model consists of an objective function that involves a parameter to control the trade-off between task completion time and total cost. They also developed evolutionary algorithms to solve the problem. However, their methods cannot guarantee the global optimality of the obtained solution. This study proposes a method to transform the nonlinear problem into a linear model and then guarantees to find a globally optimal solution of the task assignment problem.

The rest of this paper is organized as follows. Section 2 presents a review of the related research on the optimal task assignment strategy in the cloud-fog computing environment. Section 3 introduces the task assignment problem by considering the execution time and operating costs in the cloud-fog system. The proposed method is described in Section 4. Finally, Section 5 provides the conclusions.

2. Literature Review

Due to the exponential increase of data generated by network devices, the conventional cloud computing architecture cannot meet the level of low latency and quick response required by IoT applications. Therefore, fog computing has received increasing attention in recent years. Cisco first proposed fog computing that can transform network edge devices into parts of a distributed computing infrastructure for supporting IoT applications [3]. Shi et al. [7] investigated the fundamental characteristics of fog computing for healthcare systems. Yi et al. [8] discussed the definition of fog computing and similar concepts, introduced three representative applications, and identified various issues when designing and implementing fog computing systems. Yousefpour et al. [9] developed a fog-node policy that considered queue lengths and various types of requests with different processing times to minimize service delays for IoT nodes. Lee et al. [10] explored the security threats and privacy issues for implementing the IoT in a fog computing environment. Hong et al. [11] proposed an architecture to deploy IoT applications across various devices, from the edge devices to the cloud. Mahmud et al. [12] proposed a taxonomy of the fog computing environment and discussed possible challenges and features.

Cloud-fog computing allows some cloud services to be executed on the edge of the network. How to select the appropriate nodes for the tasks to be processed is critical to the performance of the cloud-fog computing architecture. Deng et al. [13] investigated a workload allocation problem considering power consumption and delay in a cloud-fog computing system. They approximately decomposed the problem into three subproblems and then solved each subproblem by existing optimization techniques. The simulations and numerical results indicated that fog computing can complement cloud computing in bandwidth conservation and transmission latency reduction. Pham and Huh [14] discussed the task scheduling problem in a cloud-fog environment. They developed a heuristic-based algorithm for task scheduling to achieve the balance between the execution time and the monetary cost of cloud resources. Nikoui et al. [15] proposed a cost-aware genetic-based task scheduling algorithm to enhance the cost efficiency for real-time applications in a fog-cloud environment. Guevara and da Fonseca [16] developed task scheduling algorithms based on integer linear programming techniques for multiclass services in cloud-fog computing systems.

Nguyen et al. [4] investigated the main techniques and the improvement criteria of the developed task assignment algorithms for cloud computing or fog computing. However, in the hybrid cloud-fog computing environment, the cloud nodes and fog nodes are different in processing capability and resource usage costs. Therefore, the tasks may not be equally assigned to all nodes [4]. The tasks should be allocated to different nodes according to the requirements of the tasks, the processing speed of nodes, and the resource usage cost of nodes. Since existing methods are not suitable for the hybrid cloud-fog computing architecture, Nguyen et al. [4] constructed a mathematical programming model to investigate the task assignment problem and developed evolutionary algorithms to solve the problem.

3. Proposed Method

Although Nguyen et al. [4] developed evolutionary algorithms to solve the task assignment problem, their methods cannot guarantee the global optimality of the obtained solution. This study transforms the nonlinear model of the task assignment problem into a linear model that is solvable by the general linear programming technique to derive a globally optimal solution.

The optimal task assignment problem in the cloud-fog computing environment discussed in this study, referring to Nguyen et al. [4], can be described as follows. Assume T_k be the kth task, then n independent tasks in $\boldsymbol{T}$ are required to be completed in the system and expressed as follows:

$$\boldsymbol{T} = \{T_1, T_2, T_3, \dots, T_n\}. \tag{1}$$

The cloud-fog computing system includes cloud nodes and fog nodes; the nodes of the same type have similar characteristics, such as CPU processing power, CPU usage cost, memory usage cost, and bandwidth usage cost. Typically, the cloud nodes have higher capabilities in computing and storage than the fog nodes, but running the tasks on the cloud nodes must pay higher costs. Assume that m nodes consisting of cloud and fog nodes in a set can be expressed as:

$$\boldsymbol{N} = \{N_1, N_2, N_3, \dots, N_m\}, \tag{2}$$

where N_i is the ith processing node. Each task T_k will be assigned to one processing node N_i, which is represented as T_k^i. Each processing node N_i $(i = 1,2,3,\dots,m)$ can be assigned multiple tasks, expressed as:

$$\boldsymbol{N_i^T} = \{T_x^i, T_y^i, \dots, T_z^i\}. \tag{3}$$

The task assignment problem considered in this study could be formulated as a node assignment of the tasks in $\boldsymbol{T}$:

$$\boldsymbol{T}^{\text{node}} = \{T_1^a, T_2^b, T_3^c, \dots, T_n^p\}. \tag{4}$$

The execution time for node N_i to complete all assigned tasks in $\boldsymbol{N_i^T}$ can be expressed as:

$$E\_Time\ (N_i) = \sum_{T_k^i \in \boldsymbol{N_i^T}} E\_Time\left(T_k^i\right) = \frac{\sum_{T_k^i \in \boldsymbol{N_i^T}} L(T_k^i)}{\text{CPUrate}(N_i)}, \tag{5}$$

where $L(T_k^i)$ is the number of instructions of task T_k^i, and $CPUrate(N_i)$ is the CPU clock rate of node N_i. $E\_Time(T_k^i) = \frac{L(T_k^i)}{\mathrm{CPUrate}(N_i)}$ is the execution time of T_k assigned in node N_i.

Assume *Timespan* is the total time to complete all tasks in T. *Timespan* can be derived by the following formula:

$$Timespan = Max_{1\le i\le m} E\_Time(N_i). \tag{6}$$

Let $Cost(T_k^i)$ be a monetary amount that must be paid for executing task T_k^i in node N_i, consisting of the processing cost $C_p(T_k^i)$, memory usage cost $C_m(T_k^i)$, and bandwidth usage cost $C_b(T_k^i)$. $Cost(T_k^i)$ can be expressed as:

$$Cost(T_k^i) = C_p(T_k^i) + C_m(T_k^i) + C_b(T_k^i). \tag{7}$$

The above three costs can be defined as:

$$C_p(T_k^i) = cost\_p_i \times E\_Time(T_k^i), \tag{8}$$

$$C_m(T_k^i) = cost\_m_i \times Memory(T_k^i), \tag{9}$$

$$C_b(T_k^i) = cost\_b_i \times Bandwidth(T_k^i), \tag{10}$$

where $cost\_p_i$ is the usage cost of CPU per time unit in node N_i, $cost\_m_i$ is the usage cost of memory per data unit in node N_i, $Memory(T_k^i)$ is the memory required by task T_k in node N_i, $cost\_b_i$ is the usage cost of bandwidth per data unit, and $Bandwidth(T_k^i)$ is the amount of bandwidth required by transmitting task T_k to be processed in node N_i.

The total cost for all tasks to be completed in a cloud-fog system can be expressed as below:

$$Total\_Cost = \sum_{T_k^i \in \boldsymbol{T}^{\mathrm{node}}} Cost(T_k^i). \tag{11}$$

Since the optimal task assignment problem considers the execution time and operating costs, Nguyen et al. [4] used an objective function to compute the trade-off between $Timespan$ and $Total\_Cost$ as follows:

$$\mathrm{Objective} = \alpha \times Timespan + (1-\alpha) \times Total\_Cost, \tag{12}$$

where $\alpha \in [0,1]$ is the trade-off coefficient between the execution time and operating costs. If $\alpha > 0.5$, the task assignment strategy concentrates on minimizing the execution time with a higher priority than the total operating costs. If $\alpha < 0.5$, minimizing the total operating costs is more important than the execution time. The value of α depends on the amount of the budget or the level of the required response time.

Nguyen et al. [4] used evolutionary algorithms to find the optimal trade-off task assignment strategy between the execution time and operating costs. Since the heuristic approaches cannot guarantee the quality of the obtained solution, this study derived the optimal assignment strategy based on the globally optimal solution by a deterministic approach. The original mathematical programming model of the task assignment problem in a cloud-fog system can be expressed as follows [4].

Model OTA1:

$$\text{minimize} \quad \alpha \times Timespan + (1-\alpha) \times Total\_Cost \tag{13}$$

subject to:

$$\sum_{i=1}^{m} T_k^i = 1,\ 1 \le k \le n, \tag{14}$$

$$Total\_Cost = \sum_{k=1}^{n} \sum_{i=1}^{m} (T_k^i \times Cost_k^i), \tag{15}$$

$$Cost_k^i = cost\_p_i \times \frac{L(T_k^i)}{\text{CPUrate}(N_i)} + cost\_m_i \times Memory(T_k^i) + cost\_b_i \times Bandwidth(T_k^i),\ \forall T_k^i \in \boldsymbol{T}, \tag{16}$$

$$Timespan = Max_{1 \le i \le m} E\_Time(N_i), \tag{17}$$

$$E\_Time(N_i) = \sum_{T_k^i \in \boldsymbol{N}_i^T} E\_Time(T_k^i) = \frac{\sum_{T_k^i \in \boldsymbol{N}_i^T} L(T_k^i)}{\text{CPUrate}(N_i)}, \tag{18}$$

where α, $L(T_k)$, $\text{CPUrate}(N_i)$, $E\_Time(T_k^i)$, $Memory(T_k^i)$, $Bandwidth(T_k^i)$, $cost\_p_i$, $cost\_m_i$, and $cost\_b_i$ are the same as described before. The decision variables are T_k^i and $T_k^i \in \{0,1\}$. Constraint (14) means that the task T_k must be assigned to only one node for execution.

After linearly expressing the constraint (17) in the above model OTA1, the original model can be transformed as follows.

Model OTA2:

$$\text{minimize} \quad \alpha \times Timespan + (1 - \alpha) \times Total\_Cost \tag{19}$$

subject to:

$$\sum_{i=1}^{m} T_k^i = 1,\ 1 \le k \le n, \tag{20}$$

$$Total\_Cost = \sum_{k=1}^{n} \sum_{i=1}^{m} (T_k^i \times Cost_k^i), \tag{21}$$

$$Cost_k^i = cost\_p_i \times \frac{T_k^i \times L(T_k^i)}{\text{CPUrate}(N_i)} + cost\_m_i \times Memory(T_k^i) + cost\_b_i \times Bandwidth(T_k^i),\ \forall T_k^i \in \boldsymbol{T}, \tag{22}$$

$$E\_Time(N_i) \le Timespan,\ \forall i, 1 \le i \le m, \tag{23}$$

$$E\_Time(N_i) = \frac{\sum_{k=1}^{n} (T_k^i \times L(T_k^i))}{\text{CPUrate}(N_i)}, \tag{24}$$

$$T_k^i \in \{0,1\}. \tag{25}$$

Model OTA2 is a mixed-integer linear model that can be solved by the optimization solver to obtain a globally optimal solution. The obtained solution may be different when the trade-off coefficient α changes.

4. Numerical Experiments and Results

This study discusses the optimal task assignment problem in a cloud-fog computing environment; several numerical experiments are presented to demonstrate the effectiveness of the proposed method. The experiments were conducted on a Notebook with a 2.6 GHz Intel Core i5-7300 CPU and 24 GB memory. All reformulated models were solved by a mathematical programming solver GUROBI 9.1.1 with default settings.

In a cloud-fog computing environment, each node has its own processing capacity, memory, and bandwidth usage cost. This study randomly generated several problems according to the parameters suggested by Nguyen et al. [4]. Table 1 lists the characteristics of the nodes used to execute the tasks in a cloud-fog computing environment. Table 2 lists the characteristics of the tasks.

Table 1. Characteristics of the nodes in a cloud-fog computing environment.

Parameter	Fog Nodes	Cloud Nodes
CPU rate (MIPS)	[500,1500]	[3000,5000]
CPU usage cost	[0.1,0.4]	[0.7,1.0]
Memory usage cost	[0.01,0.03]	[0.02,0.05]
Bandwidth usage cost	[0.01,0.02]	[0.05,0.1]

Table 2. Characteristics of the tasks to be assigned.

Property	Value
Number of instructions (10^9 instructions)	[1,100]
Memory required (MB)	[50,200]
Input file size (MB)	[10,100]
Output file size (MB)	[10,100]

In our experiments, $\alpha = 0.5$ is adopted, which means that the time and cost have identical priorities in the objective. Three datasets are used in our experiments. Referring to the research of Nguyen et al. [4], datasets 1 includes three cloud nodes and 10 fog nodes. To explore the impact of different numbers of cloud and fog nodes on the solution speed of the proposed method, dataset 2 includes five cloud nodes and 12 fog nodes, and dataset 3 includes seven cloud nodes and 14 fog nodes. Cloud nodes are more powerful for processing tasks, but the cost of using them is higher. In each dataset, 10–50 tasks are assigned to different nodes according to the globally optimal solution of Model OTA2 solved by GUROBI. For each case, ten instances with identical numbers of cloud nodes, fog nodes, and tasks are randomly generated. The average CPU time is the average running time of GUROBI to solve each instance.

As seen in Tables 3–5, the average CPU time increases as the number of tasks increases under identical numbers of cloud nodes and fog nodes. For small cases, the results can be obtained within several seconds. Comparing the average CPU time for solving the case with the same number of tasks in the three datasets, the difference is more significant as the number of cloud nodes and fog nodes increases. Since the number of cloud nodes, fog nodes, and tasks determines the number of binary variables in Model OTA2, more cloud nodes, fog nodes, or tasks results in more CPU time for solving the task assignment problem. Since Model OTA2 involves an SOS1 constraint, the technique for treating the SOS1 constraint with fewer binary variables can be considered to improve the computational efficiency of the large-scale task assignment problems in a cloud-fog computing environment.

Table 3. Experimental results of the proposed method for 3 cloud nodes and 10 fog nodes (dataset 1).

Case No.	1	2	3	4	5	6	7	8	9
Number of cloud nodes	3	3	3	3	3	3	3	3	3
Number of fog nodes	10	10	10	10	10	10	10	10	10
Number of tasks	10	15	20	25	30	35	40	45	50
Number of 0-1 variables	130	195	260	325	390	455	520	585	650
Number of continuous variables	15	15	15	15	15	15	15	15	15
Number of constraints	37	42	47	52	57	62	67	72	77
Average $Timespan$ (sec)	40.4057	63.6957	89.3140	99.3824	124.0032	107.3536	186.0245	160.2466	195.3863
Average $Total\_Cost$	145.7769	228.4093	266.5487	386.4935	498.4282	561.9562	607.3436	673.5105	764.3977
Average CPU time	0.2191	1.1957	12.0210	46.0756	102.5049	414.2663	580.9763	590.1869	677.3602

Table 4. Experimental results of the proposed method for 5 cloud nodes and 12 fog nodes (dataset 2).

Case No.	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
Number of cloud nodes	5	5	5	5	5	5	5	5	5
Number of fog nodes	12	12	12	12	12	12	12	12	12
Number of tasks	10	15	20	25	30	35	40	45	50
Number of 0-1 variables	170	255	340	425	510	595	680	765	850
Number of continuous variables	19	19	19	19	19	19	19	19	19
Number of constraints	45	50	55	60	65	70	75	80	85
Average $Timespan$ (sec)	48.9281	46.6694	59.0551	81.3217	101.6427	124.4561	141.8461	166.3881	214.1368
Average $Total\_Cost$	142.4031	239.3706	239.2414	354.4158	442.8219	535.4207	596.2381	691.7541	760.2561
Average CPU time	0.3900	1.3135	12.6975	83.7108	149.1998	453.4950	665.6732	721.8558	844.8545

Table 5. Experimental results of the proposed method for 7 cloud nodes and 14 fog nodes (dataset 3).

Case No.	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**
Number of cloud nodes	7	7	7	7	7	7	7	7	7
Number of fog nodes	14	14	14	14	14	14	14	14	14
Number of tasks	10	15	20	25	30	35	40	45	50
Number of 0-1 variables	210	315	420	525	630	735	840	945	1050
Number of continuous variables	23	23	23	23	23	23	23	23	23
Number of constraints	53	58	63	68	73	78	83	88	93
Average $Timespan$ (sec)	45.4530	60.9926	79.7936	91.4356	115.0861	103.8784	130.1187	130.7363	164.7988
Average $Total\_Cost$	125.9765	195.1031	270.6764	324.1400	421.4838	494.4605	609.9551	679.6400	729.3251
Average CPU time	0.4890	1.3852	119.1511	148.9607	470.8410	679.0558	706.5134	887.7305	1157.8949

5. Conclusions

In the era of IoT, how to improve the quality of service in IoT networks becomes a challenging problem. Since conventional cloud computing architectures do not meet the requirements of the IoT applications, edge computing and fog computing have attracted increasing attention from the industrial and academic sectors in recent years. Although evolutionary algorithms can solve the task assignment problem with hundreds of tasks in the cloud-fog system, the quality of the obtained solution cannot be guaranteed. This study developed a linearization method to solve the nonlinear task assignment problem. Therefore, the global optimality of the obtained solution can be guaranteed by using the proposed deterministic optimization approach. This proposed method can allocate tasks to different cloud nodes or fog nodes based on the requirements of the tasks, the processing speed of nodes, and the resource usage cost of nodes in a cloud-fog system.

Although the deterministic method can find a global optimum, the limitation of the proposed approach is that the computational complexity grows rapidly as the problem size increases. More investigation and research will be required to develop an efficient approach for solving the task assignment problem in a cloud-fog system.

Author Contributions: Conceptualization, M.-H.L., J.-F.T., and C.-H.H.; methodology, M.-H.L.; software, M.-H.L.; validation, J.-F.T. and M.-H.L.; formal analysis, J.-F.T.; investigation, J.-F.T. and C.-H.H.; resources, M.-H.L. and C.-H.H.; data curation, M.-H.L.; writing—original draft preparation, M.-H.L.; writing—review and editing, M.-H.L., C.-H.H. and J.-F.T.; visualization, M.-H.L.; supervision, M.-H.L. and J.-F.T.; project administration, M.-H.L. and J.-F.T.; and funding acquisition, M.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Ministry of Science and Technology in Taiwan under Grants MOST 108-2410-H-027-018, MOST 109-2410-H-027-012-MY2 and MOST 108-2410-H-845-028.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. IoT Business News, 2020, Global IoT Device Connections to Reach 11.7 billion in 2020, Surpassing non-IoT Devices for the First time. Available online: https://iotbusinessnews.com/2020/11/20/03121-global-iot-device-connections-to-reach-11-7-billion-in-2020-surpassing-non-iot-devices-for-the-first-time/ (accessed on 20 November 2020).
2. International Data Corporation, 2020. IoT Growth Demands Rethink of Long-Term Storage Strategies. Available online: https://www.eetasia.com/iot-growth-demands-rethink-of-long-term-storage-strategies/ (accessed on 29 July 2020).
3. Cisco White Paper, 2015. Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. Available online: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf (accessed on 20 November 2020).
4. Nguyen, B.M.; Binh, H.T.T.; Anh, T.T.; Son, D.B. Evolutionary Algorithms to Optimize Task Scheduling Problem for the IoT Based Bag-of-Tasks Application in Cloud–Fog Computing Environment. *Appl. Sci.* **2019**, *9*, 1730, doi:10.3390/app9091730.
5. Pham, X.-Q.; Man, N.D.; Tri, N.D.T.; Thai, N.Q.; Huh, E.-N. A cost- and performance-effective approach for task scheduling based on collaboration between cloud and fog computing. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, doi:10.1177/1550147717742073.
6. Klonoff, D.C. Fog Computing and Edge Computing Architectures for Processing Data From Diabetes Devices Connected to the Medical Internet of Things. *J. Diabetes Sci. Technol.* **2017**, *11*, 647–652, doi:10.1177/1932296817717007.
7. Shi, Y.; Ding, G.; Wang, H.; Roman, H.E.; Lu, S. The fog computing service for healthcare. In Proceedings of the 2015 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), Beijing, China, 28–30 May 2015; pp. 1–5.
8. Yi, S.; Li, C.; Li, Q. A survey of fog computing: Concepts, applications and issues. In Proceedings of the 2015 Workshop on Mobile Big Data, Santa Clara, CA, USA, 29 October–1 November 2015; pp. 37–42.
9. Yousefpour, A.; Ishigaki, G.; Jue, J.P. Fog Computing: Towards Minimizing Delay in the Internet of Things. In Proceedings of the 2017 IEEE International Conference on Edge Computing (EDGE), Honolulu, HI, USA, 25-30 June 2017; IEEE, 2017; pp. 17–24.
10. Lee, K.; Kim, D.; Ha, D.; Rajput, U.; Oh, H. On security and privacy issues of fog computing supported Internet of Things environment. In Proceedings of the 2015 6th International Conference on the Network of the Future (NOF), Montreal, Quebec, Canada, 30 September-2 October, 2015; IEEE, 2015; pp. 1–3.
11. Hong, K.; Lillethun, D.; Ramachandran, U.; Ottenwälder, B.; Koldehofe, B. Mobile fog: A programming model for large-scale applications on the internet of things. In Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing, Hong Kong, China, 2013; pp. 15–20.
12. Mahmud, R.; Kotagiri, R.; Buyya, R. Fog Computing: A Taxonomy, Survey and Future Directions. In *Internet of Things*; Springer, Singapore, , 2018; pp. 103–130.
13. Deng, R.; Lu, R.; Lai, C.; Luan, T.H. Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing. In Proceedings of the 2015 IEEE international conference on communications, London, UK, 2015; pp. 3909–3914.
14. Pham, X.-Q.; Huh, E.-N. Towards task scheduling in a cloud-fog computing system. In Proceedings of the 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS), Kanazawa, Japan, 5-7 October 2016; IEEE, 2016; pp. 1–4.
15. Nikoui, T.S.; Balador, A.; Rahmani, A.M.; Bakhshi, Z. Cost-Aware Task Scheduling in Fog-Cloud Environment. In Proceedings of the 2020 CSI/CPSSI International Symposium on Real-Time and Embedded Systems and Technologies (RTEST), Tehran, Iran, 10-11 June, 2020; IEEE, 2020; pp. 1–8.
16. Guevara, J.C.; Da Fonseca, N.L.S. Task scheduling in cloud-fog computing systems. *Peer-to-Peer Netw. Appl.* **2021**, 1–16, doi:10.1007/s12083-020-01051-9.