Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20260502105
Volume 5, Issue2 (May-August 2026), PP: 964-971.
www.indjcst.com
ISSN No: 2583-5300

A Comprehensive Review of Offloading Algorithms in Replicated Fog Computing Environments

Ashish Bagla[1], Dr. Deepak Dagar[2], Dr. Pratik Shrivastava[3]

[1]*Research Scholar, Department of Computer Science and Engineering, Shri Venkateshwara University, Gajraula, Uttar Pradesh, India.*

[2]*Supervisor, Department of Computer Science and Engineering, Shri Venkateshwara University, Gajraula, Uttar Pradesh, India.*

[3]*Assistant Professor, Department of Computer Science and Engineering, Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India.*

To Cite this Article: *Ashish Bagla[1], Dr. Deepak Dagar[2], Dr. Pratik Shrivastava[3], "A Comprehensive Review of Offloading Algorithms in Replicated Fog Computing Environments", Indian Journal of Computer Science and Technology, Volume 05, Issue 02 (May-August 2026), PP: 964-971.*

Abstract: *This paper provides an in-depth review of computation offloading algorithms within replicated fog computing environments, a critical area for modern distributed systems. It delineates the foundational concepts of fog computing and data replication, exploring their architectures, characteristics, and inherent challenges. The paper then systematically examines various offloading mechanisms and objectives, categorizing state-of-the-art algorithms, including optimization-based, machine learning-driven, and dynamic resource management approaches. A significant focus is placed on the complex interplay between offloading and replication, particularly concerning data consistency, fault tolerance, resource heterogeneity, and security. Through an analysis of real-world applications and identified research gaps, this review aims to serve as a foundational resource for PhD researchers, guiding future investigations into more robust, efficient, and intelligent offloading strategies in highly dynamic and resource-constrained fog environments.*

I. INTRODUCTION

1.1. Evolution of Distributed Computing Paradigms: From Cloud to Fog and Edge

The landscape of computing has undergone a significant transformation; driven by the proliferation of Internet of Things (IoT) devices and the exponential growth of data they generate. Traditionally, centralized cloud computing offered unparalleled scalability and flexibility, allowing companies and individuals to process, store, and analyse vast amounts of data on virtual servers via the internet. This model, however, began to encounter limitations as the volume and velocity of data increased, leading to issues such as high latency, network congestion, and bandwidth constraints when data had to traverse long distances to centralized data centres.

In response to these challenges, new distributed computing paradigms emerged, notably fog computing and edge computing. Fog computing was introduced as a decentralized computing architecture designed to bridge the gap between resource-constrained edge devices and distant cloud data centers. Positioned as an intermediate layer, fog computing processes data closer to its source, thereby reducing the strain on the core network and improving responsiveness. Edge computing, on the other hand, focuses on processing data directly at or very near the data source, offering the lowest possible latency for time-critical applications. Fog computing extends the capabilities of edge computing by providing a more hierarchical and distributed approach, often described as "cloud closer to the ground".

1.2 The Role and Benefits of Data Replication in Distributed Systems

Data replication is a fundamental process in distributed systems, involving the copying and maintenance of data across multiple physical or logical locations. Its primary purpose is to ensure consistency, availability, and reliability of data, especially in environments where single points of failure could compromise system operations.

The benefits of data replication are extensive and critical for robust distributed systems:

- Improved Availability
- Disaster Recovery
- Load Balancing
- Minimized Latency
- Enhanced Reliability and Fault Tolerance
- Scalability

1.3 Defining the Research Landscape: Offloading in Replicated Fog Computing

The convergence of computation offloading and replicated fog computing environments represents a complex yet highly promising area of research. Fog computing, with its distributed, heterogeneous, and dynamic nodes, presents a unique set of challenges for task allocation and resource management. While replication offers significant benefits in terms of availability and fault tolerance, its implementation within a highly distributed and dynamic fog environment introduces its own complexities, including maintaining data consistency, managing synchronization overhead, and mitigating potential data inconsistencies.

The research in this domain focuses on designing intelligent offloading algorithms that can effectively leverage replicated resources within the fog. The goal is to optimize various performance metrics while simultaneously ensuring data integrity and overall system reliability. This intersection is inherently multi-dimensional. Offloading is complex on its own, involving decisions of "what," "where," "when," and "how" to offload. Replication is also complex, dealing with consistency models, synchronization, and fault tolerance. When these two concepts are combined in a dynamic, heterogeneous fog environment, their complexities multiply. For instance, offloading a task to a replicated fog node requires not only deciding which node is optimal but also which specific replica to utilize and how to ensure consistency across all other replicas, especially if the task modifies shared data. This creates a multi-dimensional optimization problem that traditional methods often struggle to address comprehensively. Consequently, this domain inherently demands multi-objective and sophisticated algorithmic solutions that can reason about computational, communication, and data management trade-offs simultaneously.

1.4 Paper Structure

The remainder of this paper is structured to provide a comprehensive review of offloading algorithms in replicated fog computing environments. Section 2 delves into the foundational concepts of fog computing and data replication, detailing their architectures, characteristics, and underlying principles. Section 3 elaborates on the mechanisms and objectives of computation offloading, classifying various strategies. Section 4 presents a detailed overview of state-of-the-art offloading algorithms, including optimization-based, machine learning-driven, and dynamic resource management approaches, with a particular focus on their applicability in replicated fog environments. Section 5 addresses the critical challenges and considerations inherent in this domain, such as data consistency, fault tolerance, and resource heterogeneity. Section 6 explores real-world applications and use cases that benefit from these advancements. Finally, Section 7 identifies key research gaps and outlines future directions for investigation, followed by a concluding summary in Section 8.

II.FOUNDATIONAL CONCEPTS: FOG COMPUTING AND DATA REPLICATION

2.1 Fog Computing: Architecture, Characteristics, and Distinctions

Fog computing is a decentralized computing architecture that extends the capabilities of cloud computing closer to the data sources, positioning itself as an intermediary layer between edge devices and the centralized cloud. This architectural model is designed to address the limitations of traditional cloud-centric approaches, particularly in scenarios demanding low latency and real-time data processing.

Key characteristics define fog computing:

- Decentralized
- Resource-constrained (relative to cloud)
- Physical Proximity and Location Awareness
- Real-time Data Processing
- Local Computation and Storage
- Reduced Bandwidth Requirements
- Enhanced Privacy
- Heterogeneity
- Dynamic Nature

The distinctions between cloud, fog, and edge computing are summarized in the table below:

2.2 Data Replication in Distributed Systems: Principles and Models

Data replication is a cornerstone of robust distributed systems, involving the creation and maintenance of multiple copies of data across various nodes or locations. This fundamental technique is employed to achieve critical system properties such as consistency, availability, and reliability.

2.2.1 Objectives and Advantages of Data Replication

The primary objectives driving the adoption of data replication are:

- Improved Availability
- Disaster Recovery
- Load Balancing
- Minimized Latency
- Enhanced Reliability and Fault Tolerance
- Scalability

Given the dynamic, heterogeneous, and potentially unreliable nature of individual fog nodes, replication moves from being a mere performance or availability enhancement to a fundamental requirement for dependable service delivery. The benefits of replication directly counter many of the challenges posed by distributed fog nodes, such as unpredictability and resource constraints.

2.2.2 Types and Schemes of Data Replication

Data replication can be categorized into various types and schemes, each suited for different system requirements and trade-offs.

Types of Data Replication:

- Transactional Replication
- Snapshot Replication
- Merge Replication

Data Replication Schemes:

- Full Replication
- Partial Replication
- No Replication

The choice between full and partial replication directly impacts the overhead (storage, bandwidth, synchronization) and the level of granularity at which consistency must be managed. In resource-constrained fog environments, partial replication might be more feasible to conserve resources, but it introduces complexity in determining what specific data to replicate and managing data consistency across different partial views. This decision is not trivial and often depends on the application's specific data access patterns and consistency requirements. Offloading algorithms in replicated fog environments need to be aware of the chosen replication scheme, as it dictates the available data, the cost of accessing it, and the potential for inconsistencies.

2.2.3Consistency Models for Replicated Data (Strong, Eventual, Causal)

Consistency models define the rules or guarantee that dictate how shared data appears to processes in a distributed system, providing a framework for reasoning about the ordering and visibility of read and write operations on replicated data. The choice of consistency model is a critical design decision, as it fundamentally impacts the trade-offs between data consistency, system performance, and availability.

- Strong Consistency
- Eventual Consistency
- Causal Consistency

The choice of consistency model is a fundamental design decision in distributed systems, directly impacting performance (latency, throughput) and reliability (data accuracy). In the context of fog computing, where low latency is a primary objective, opting for strong consistency can negate the performance benefits of offloading by introducing significant synchronization overhead.

Synchronous Replication commits updates to all replicas before acknowledging the write, ensuring strong consistency but increasing latency. **Asynchronous Replication** acknowledges the write before updates propagate to all replicas, reducing latency but potentially leading to eventual consistency.

III.COMPUTATION OFFLOADING: MECHANISMS AND OBJECTIVES

3.1Core Principles and Decision-Making in Computation Offloading

Computation offloading is a pivotal technique in modern distributed computing, fundamentally involving the transfer of computationally intensive tasks from resource-limited devices to more powerful remote processing nodes. This practice is essential for overcoming hardware limitations, such as restricted computational power, storage, and energy, inherent in devices like smartphones and IoT sensors.

The offloading process is typically conceptualized as a series of four sequential steps: task generation, task allocation, communication and computing, and decision making. At the heart of effective offloading lies a set of critical decisions that determine its efficiency and impact. These decisions are often framed around the fundamental questions of "what," "where," "when," and "how" to offload.

- What to Offload
- Where to Execute
- When to Offload
- How to Offload

The numerous decision points (what, where, when, how) reveal that offloading is not a simple binary choice but a complex multi-dimensional optimization problem. Each decision point, such as choosing between full or partial offloading, or between local, fog, or cloud execution, introduces a new set of trade-offs and dependencies. For instance, deciding *what* to offload (e.g., splitting a task) directly impacts the *how* (communication overhead for sub-tasks) and *where* (different tiers might be optimal for different sub-tasks).

3.2 Offloading Objectives: Energy Efficiency, Latency Reduction, Cost Optimization, and QoS Enhancement

Computation offloading is driven by a diverse set of objectives, which often need to be optimized simultaneously, despite their frequently conflicting nature. The primary goals of offloading mechanisms in fog computing environments include:

- Energy Efficiency
- Latency Reduction
- Cost Optimization
- Quality of Service (QoS) Enhancement
- Load Balancing

The numerous objectives listed (energy, latency, cost, QoS, load balancing) are often in conflict. For example, minimizing energy consumption might mean offloading all tasks, which could inadvertently increase latency due to transmission time and queueing delays at the remote server. Conversely, prioritizing ultra-low latency might necessitate local processing or offloading to very close, potentially less powerful, nodes, which could increase local energy consumption or lead to resource contention.

3.3Classification of Offloading Strategies: Full vs. Partial, Horizontal vs. Vertical

Offloading strategies can be broadly classified based on the granularity of the tasks being offloaded and the direction of the offloading within a multi-tier computing environment. These classifications are crucial for understanding the design space of offloading algorithms.

- Full (Binary) Offloading
- Partial Offloading
- Offloading Directions (in Hierarchical Fog Environments
  - Horizontal Offloading
  - Vertical Offloading
  - Hybrid Offloading

The distinction between full and partial offloading, combined with horizontal and vertical directions, defines the adaptability of the offloading strategy. Partial offloading, while more complex, allows for finer-grained resource utilization and potentially better optimization, especially in heterogeneous fog environments where different components of a task might be best suited for different nodes. Horizontal offloading is crucial for load balancing within a tier, while vertical offloading enables scaling to more powerful resources when local capacity is insufficient.

IV.OFFLOADING ALGORITHMS IN REPLICATED FOG ENVIRONMENTS: STATE-OF-THE-ART

The design of efficient offloading algorithms in replicated fog environments is a critical area of research, driven by the need to optimize performance metrics while ensuring reliability and data consistency across distributed, heterogeneous, and dynamic nodes. State-of-the-art approaches primarily fall into optimization-based methods, machine learning and deep reinforcement learning (ML/DRL) techniques, and dynamic resource management strategies.

4.1Optimization-Based Approaches

- Game Theory-Driven Algorithms for Resource Allocation and Offloading Decisions
- Meta-heuristic Algorithms (e.g., PSO, GA, GWO, ACO) for Task Scheduling
- Incentive Mechanisms for High-Cost Task Offloading

4.2Machine Learning and Deep Reinforcement Learning (ML/DRL) Approaches

Machine Learning (ML) and Deep Reinforcement Learning (DRL) algorithms have emerged as powerful paradigms for addressing the complexities of computation offloading in fog computing environments. These approaches are increasingly considered efficient alternatives to traditional optimization strategies, enabling more intelligent and adaptive decisions for task offloading and resource allocation.

4.2.1Taxonomy of ML/DRL Algorithms in Fog Offloading

ML/DRL algorithms applied in fog offloading can be broadly categorized into three main types based on their underlying learning mechanisms:

- Value-Based Algorithms
- Policy-Based Algorithms
- Hybrid Algorithms.

The following table provides a taxonomy of RL/DRL algorithms for fog offloading:

Category	Specific Algorithms	Key Principle	Common Applications in Fog Offloading	Noted Limitations (in some studies)
	Q-Learning	Model-free, learns optimal action based on Q-value function through trial and error.	Reduce total latency, load balancing, energy consumption, QoS, resource utilization.	Mobility, security, communication costs, scalability not always considered.

Value-Based	Deep Q Network (DQN)	Uses neural network to estimate Q-values, addresses Q-learning's tabular approach.	Computation offloading, resource allocation, delay minimization, throughput maximization, optimal execution locations.	Mobility, security not always considered.
	Double DQN (DDQN)	Addresses overestimation in DQN by splitting evaluation and selection.	Reduce costs (execution delay, power consumption), minimize overall energy usage, high-dimensional state spaces.	Security, mobility not always considered.
	Dueling DQN	Separates state-value and advantage functions for improved performance.	Select suitable offloading strategy for UEs, optimize overall utility in F-RANs.	Mobility, security not always considered.
	Deep Recurrent Q-Network (DRQN)	Enhances DQN with LSTM for memorizing actions and handling partial observability.	Estimate optimal value functions, ideal offloading policies for IoT, minimize energy under delay constraints, enhance resource utilization.	Security, mobility not always considered.
	SARSA	On-policy temporal difference learning, selects actions while learning a policy.	Fuzzy reinforcement learning for energy-saving task offloading in vehicular fog.	Security, mobility not always considered.
Policy-Based	Policy Gradient (PG)	Optimizes parameterized policies using gradient descent to maximize expected returns.	Computation offloading in IoT fog systems with energy harvesting, maximize latency-satisfied utility.	Security not always considered.
	Advantage Actor-Critic (A2C)	Concurrently predicts an actor (policy) and a critic (value function).	Joint optimization of resource allocation and offloading strategies to decrease delay, collaborative content caching/resource allocation.	Security not always considered.
	Proximal Policy Optimization (PPO)	Extension of A2C with a clipped objective function to prevent sudden policy changes.	Assign limited fog resources to vehicular applications (minimize service delays), UAV-assisted Offloading (reduce queue length, energy).	Energy of vehicles, security, mobility not always considered.
Hybrid	Deep Deterministic Policy Gradient (DDPG)	Combines DQN and DPG; uses primary/target networks with actor/critic.	Optimize offloading computations in fog, enhance cooperative processing efficiency, improve user service experience, joint task partitioning/power control.	Security, privacy, mobility, scalability not always considered.
	Soft Actor-Critic (SAC)	Off-policy A2C model with sample-efficient learning and entropy/stability optimization.	Handle task allocation problems in dynamic vehicular environments, V2V partial computation offloading.	Security not always considered.

4.2.2 Applications of ML/DRL in Dynamic Offloading and Resource Management

These techniques are increasingly applied to optimize various aspects of offloading and resource management:

- Optimal Offloading Decisions and Resource Allocation
- Dynamic Task Offloading Frameworks
- Fault-Tolerant Scheduling
- Dynamic Resource Management

4.2.3 Emerging Trends: Federated Learning and Digital Twin Integration

The field of offloading algorithms in fog computing is continuously evolving, with the integration of advanced concepts like Federated Learning and Digital Twins representing significant emerging trends. These integrations signify a deeper convergence of cutting-edge AI concepts with distributed systems principles.

- Federated Learning (FL) Integration
- Digital Twin (DT) Integration

4.3 Dynamic Resource Management and Load Balancing Strategies

These strategies aim to distribute incoming requests and computational loads effectively among various fog nodes, preventing any single node from becoming overwhelmed and ensuring efficient resource utilization.

Key strategies and approaches for dynamic resource management and load balancing include:

- Dynamic Threshold-based Resource Management (DTRM
- Community-based Resource Allocation
- Intelligent Energy Perception Schemes
- Orchestrator Replication for On-Demand Clusters
- Addressing Challenges in Dynamic Environments

V.CRITICAL CHALLENGES AND CONSIDERATIONS

Despite the significant advancements in offloading algorithms within replicated fog computing environments, several critical challenges and considerations persist. These issues stem from the inherent distributed, heterogeneous, and dynamic nature of fog infrastructures, often compounded by the complexities introduced by data replication.

- Maintaining Data Consistency in Replicated Fog Environments
- Ensuring Fault Tolerance and Reliability
- Managing Resource Heterogeneity and Dynamic Workloads
- Addressing Security and Privacy Concerns
- Scalability and Network Congestion

VI.REAL-WORLD APPLICATIONS AND USE CASES

The unique capabilities of fog computing, particularly its ability to provide low latency, real-time processing, and localized data handling, make it an ideal solution for a wide array of real-world applications. When combined with replication, these applications gain enhanced reliability, availability, and fault tolerance, enabling robust operation in dynamic and critical environments.

- Internet of Things (IoT) and Smart Systems (Homes, Cities, Healthcare)
- Vehicular Fog Computing (VFC) and Autonomous Systems
- Industrial Automation and Cloud Robotics

VII.RESEARCH GAPS AND FUTURE DIRECTIONS

Despite the significant progress in offloading algorithms within replicated fog computing environments, several critical research gaps and open issues remain. Addressing these challenges is essential for the widespread adoption and full realization of fog computing's potential.

7.1 Advanced Consistency Models for Dynamic Fog Replication

Current consistency models, whether strong or eventual, present inherent trade-offs between consistency guarantees and performance (latency, throughput).[23] In highly dynamic and heterogeneous fog environments, a single, static consistency model is often insufficient. Research is needed to develop more advanced and adaptive consistency models that can dynamically adjust based on application requirements, network conditions, and the criticality of data. This includes exploring:

- Context-Aware Consistency
- Hybrid Consistency Models
- Consistency-as-a-Service
- Quantifying Consistency Trade-offs

7.2 Enhanced Fault Tolerance Mechanisms for Unpredictable Environments

The unpredictable nature of fog nodes, including their mobility, intermittent connectivity, and potential for sudden failures, poses significant challenges for ensuring continuous service availability and data integrity. While replication is a foundational mechanism, existing fault tolerance strategies require enhancement:

- Proactive and Predictive Fault Management
- Adaptive Fault Tolerance Strategies
- Mobility-Aware Fault Tolerance
- Resource-Constrained Fault Tolerance

7.3 Synergistic Integration of Emerging Technologies (e.g., Blockchain, Digital Twins, FL)

The integration of emerging technologies holds immense promise for overcoming current limitations and unlocking new capabilities in fog computing:

- Blockchain for Trust and Security
- Digital Twins for Predictive Optimization
- Federated Learning for Privacy-Preserving AI
- Quantum Computing Integration

7.4 Standardization and Interoperability Across Heterogeneous Fog Deployments

The lack of universally accepted standards for fog computing is a significant hurdle, leading to compatibility issues between different fog systems and services. The inherent heterogeneity of fog nodes and network infrastructures further complicates interoperability. Future research and industry efforts must focus on:

- Standardized APIs and Protocols
- Open-Source Frameworks and Platforms
- Federation Models

7.5 Holistic Optimization for Multi-Objective Scenarios

As highlighted, offloading optimization inherently involves multiple, often conflicting, objectives (e.g., minimizing latency, energy consumption, and cost while maximizing QoS and reliability). Current research often focuses on optimizing a subset of these objectives or finding trade-offs. Future directions should explore:

- Truly Multi-Objective Optimization Algorithms.
- Dynamic Weighting of Objectives
- Cross-Layer Optimization

7.6 Development of Realistic Testbeds and Simulation Tools

The evaluation of proposed offloading algorithms often relies heavily on simulations due to the practical infeasibility and high cost of conducting large-scale real-world experiments in dynamic fog environments. However, existing simulation tools have limitations:

- Realism and Fidelity
- Support for New Features
- Comparative Analysis Challenges
- Development of Physical Testbeds

VIII.CONCLUSION

The domain of offloading algorithms in replicated fog computing environments represents a critical frontier in distributed systems research, driven by the escalating demands of IoT applications and the inherent limitations of traditional cloud-centric models. This paper has systematically reviewed the foundational concepts, state-of-the-art algorithmic approaches, and persistent challenges within this complex interplay.

Fog computing emerges as a vital intermediary layer, strategically positioned to provide localized processing, reduced latency, and enhanced privacy, thereby complementing rather than replacing cloud computing. Its decentralized, heterogeneous, and dynamic nature, however, introduces significant complexities that necessitate sophisticated offloading strategies. Data replication, while fundamental for ensuring availability, reliability, and fault tolerance in distributed systems, further compounds these complexities, particularly concerning data consistency management in highly dynamic fog environments. The inherent trade-off between strong consistency and low latency remains a central design dilemma, compelling the adoption of adaptive or tunable consistency models.

State-of-the-art offloading algorithms are increasingly leveraging advanced optimization techniques, including game theory for decentralized decision-making and meta-heuristics for solving NP-hard task scheduling problems. A significant paradigm shift is observed with the widespread adoption of Machine Learning and Deep Reinforcement Learning (ML/DRL) approaches. These AI-driven algorithms are uniquely capable of learning optimal, adaptive policies in real-time, addressing the unpredictability and heterogeneity of fog environments. Emerging trends such as Federated Learning and Digital Twin integration promise to further revolutionize offloading by enabling privacy-preserving distributed AI and proactive, predictive resource management, respectively. Dynamic resource management and load balancing strategies are essential to continuously adapt to fluctuating workloads and ensure efficient utilization of diverse fog resources.

Despite these advancements, critical challenges persist. Maintaining data consistency across replicated fog nodes, ensuring robust fault tolerance in unpredictable environments, effectively managing resource heterogeneity and dynamic workloads, and addressing evolving security and privacy concerns remain formidable tasks. Furthermore, achieving true scalability and mitigating network congestion in vast, geographically distributed fog deployments requires continuous innovation.

The research landscape points towards a future where offloading algorithms are not merely about task migration but about intelligent, autonomous, and holistic resource orchestration across the entire cloud-to-thing continuum. Future research must prioritize the development of advanced, context-aware consistency models, enhanced mobility-aware fault tolerance mechanisms, and the synergistic integration of emerging technologies like blockchain and quantum computing. The establishment of standardized protocols and the development of realistic testbeds are also crucial for validating theoretical advancements and fostering widespread adoption. By addressing these intricate challenges, the field can unlock the full potential of replicated fog computing, paving the way for more resilient, efficient, and intelligent distributed systems that can truly empower the next generation of real-world applications in smart cities, healthcare, autonomous systems, and industrial automation.

REFERENCES

1. Meng, L., et al. (2023). Task offloading optimization mechanism based on deep neural network in edge-cloud environment. Journal of Cloud Computing, 12, Article 76.
2. Tang, L., et al. (2021). Joint optimization of network selection and task offloading for vehicular fog computing. IEEE Internet of Things Journal, 8(5), 10376-10389.
3. Pakmehr, S., et al. (2024). Exploring future directions for fault-tolerant strategies in replicated fog computing environments. ACM Computing Surveys, 56(2), 457-490.
4. A. Bhattacharya and P. De, ``A survey of adaptation techniques in computationoffloading,'' J. Netw. Comput. Appl., vol. 78, pp. 97115, Jan. 2017, doi: 10.1016/j.jnca.2016.10.023.
5. K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, ``A survey of computation offloading for mobile systems,'' Mobile Netw. Appl., vol. 18, no. 1, pp. 129140, Apr. 2012, doi: 10.1007/s11036-012-0368-0.
6. A. Yousefpour, G. Ishigaki, and J. P. Jue, ``Fog computing: Towards minimizing delay in the Internet of Things,'' in Proc. IEEE Int. Conf. Edge Comput. (EDGE), Jun. 2017, pp. 1724.
7. L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, ``Cost efcient resource management in fog computing supported medical cyber-physical system,'' IEEE Trans. Emerg. Topics Comput., vol. 5, no. 1, pp. 108119, Jan. 2017.
8. R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, ``Optimal workload allocation in fog-cloud computing towards balanced delay and power consumption,'' IEEE Internet Things J., vol. 3, no. 6, pp. 11711181, Dec. 2016.
9. D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, ``Joint optimization of task scheduling and image placement in fog computing supported software defined embedded system,'' IEEE Trans. Comput., vol. 65, no. 12, pp. 37023712, Dec. 2016.
10. Y.-C. Chen, Y.-C. Chang, C.-H. Chen, Y.-S. Lin, J.-L. Chen, and Y.-Y. Chang, ``Cloud-fog computing for information-centric Internetof-Things applications,'' in Proc. Int. Conf. Appl. Syst. Innov. (ICASI), May 2017, pp. 637640.
11. Y.-L. Jiang, Y.-S. Chen, S.-W. Yang, and C.-H. Wu, ``Energy-efficient task offloading for time-sensitive applications in fog computing,'' IEEE Syst. J., vol. 13, no. 3, pp. 29302941, Sep. 2019.
12. M. Huang,W. Liu, T. Wang, A. Liu, and S. Zhang, ``A cloud-MEC collaborative task offloading scheme with service orchestration,'' IEEE Internet Things J., to be published.
13. Y. Liu, Z. Zeng, X. Liu, X. Zhu, and M. Z. A. Bhuiyan, ``A novel load balancing and lowresponse delay framework for edge-cloud network based on SDN,'' IEEE Internet Things J., to be published.
14. S. Zahoor, N. Javaid, A. Khan, B. Ruqia, F. J. Muhammad, and M. Zahid, ``A Cloud-Fog-Based smart grid model for efcient resource utilization,'' in Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC), Jun. 2018, pp. 11541160.
15. S. Zahoor, S. Javaid, N. Javaid, M. Ashraf, F. Ishmanov, and M. Afzal, ``Cloud-fog-based smart grid model for efficient resource management,'' Sustainability, vol. 10, no. 6, p. 2079, Jun. 2018. [Online]. Available: https://www.mdpi.com/2071-1050/10/6/2079
16. S. A. A. Naqvi, N. Javaid, H. Butt, M. B. Kamal, A. Hamza, and M. Kashif, ``Metaheuristic optimization technique for load balancing in cloud-fog environment integrated with smart grid,'' in Advances in Network-Based Information Systems,,
17. L. Barolli, N. Kryvinska, T. Enokido, and M. Takizawa, Eds. Cham, Switzerland: Springer, 2019, pp. 700711.
18. S. Ningning, G. Chao, A. Xingshuo, and Z. Qiang, ``Fog computing dynamic load balancing mechanism based on graph repartitioning,'' China Commun., vol. 13, no. 3, pp. 156164, Mar. 2016.
19. S. Bitam, S. Zeadally, and A. Mellouk, ``Fog computing job scheduling optimization based on bees swarm,'' Enterprise Inf. Syst., vol. 12, no. 4, pp. 373397, Apr. 2017, doi: 10.1080/17517575.2017.1304579.
20. H. T. T. Binh, T. T. Anh, D. B. Son, P. A. Duc, and B. M. Nguyen, ``An evolutionary algorithm for solving task scheduling problem in cloud-fog computing environment,'' in Proc. 9th Int. Symp. Inf. Commun. Technol. (SoICT), New York, NY, USA, 2018, pp. 397404, doi: 10.1145/3287921.3287984.
21. B. M. Nguyen, H. Thi Thanh Binh, T. The Anh, and D. Bao Son, ``Evolutionary algorithms to optimize task scheduling problem for the IoT based Bag-of-Tasks application in Cloud-Fog computing environment,'' Appl. Sci., vol. 9, no. 9, p. 1730, Apr. 2019.[Online]. Available: https://www.mdpi.com/2076-3417/9/9/1730
22. A. Mebrek, L. Merghem-Boulahia, and M. Esseghir, ``Efcient green solution for a balanced energy consumption and delay in the IoT-Fog-Cloud computing,'' in Proc. IEEE 16th Int. Symp. Netw. Comput. Appl. (NCA), Oct. 2017, pp. 14.
23. Y. Li and S. Wang, ``An energy-aware edge server placement algorithm in mobile edge computing,'' in Proc. IEEE Int. Conf. Edge Comput. (EDGE), Jul. 2018, pp. 6673, doi: 10.1109/edge.2018.00016.
24. C. Canali and R. Lancellotti, ``GASP: Genetic algorithms for service placement in fog computing systems,'' Algorithms, vol. 12, no. 10, p. 201, Sep. 2019. [Online]. Available: https://www.mdpi.com/1999- 4893/12/10/2011