International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 5, Issue 4, 130-136, 2024
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V5I4P113

Original Article

Service Placement Strategies in Hybrid Cloud-Fog Architectures for Latency-Sensitive IoT Applications

Naresh Kalimuthu
Independent Researcher

Abstract *- The growth of the Internet of Things (IoT) has introduced new latency-sensitive applications that challenge traditional centralized cloud computing models. In response, fog computing has emerged as an innovative approach that extends cloud services to the network edge, enabling faster processing and improved latency. This paper reviews the placement of hybrid cloud-fog architectures and the demands of latency-sensitive IoT applications. We aim to describe and analyze three key challenges: managing resource heterogeneity, maintaining seamless service for mobile users, and resolving QoS (Quality of Service) goal conflicts. The paper systematically examines how these challenges have been addressed through heuristic algorithms, mobility-aware migration policies, and various multi-objective optimization techniques. Evidence supporting the effectiveness of these strategies comes from reviews of prominent approaches and relevant case studies.*

Keywords *- Fog Computing, Cloud Computing, Internet of Things (IoT), Service Placement, Latency-Sensitive Applications, Quality of Service (QoS), Resource Allocation.*

1. Introduction

1.1. Exponential Growth of the IoT Ecosystem

Technology has undergone significant changes recently, largely driven by the Internet of Things (IoT). Earlier forecasts predicted that billions of devices would connect and generate data, potentially reaching trillions of gigabytes to be collected. This data, created by the growing network of sensors and smart devices, drives a fundamental shift in computing models.[3] The volume and speed of data are quickly surpassing the capabilities of traditional data processing and storage systems, which is why new data architecture frameworks are necessary.

1.2. Constraints of Cloud Platform

The centralized cloud computing model is a standard framework that has provided on-demand computation and storage for many years. The architecture involving geo-distributed massive data has inherent limitations when it comes to supporting growing classes of high-priority IoT applications. These challenges include the gap between demand and network bandwidth, which is extremely valuable. The physical distance between IoT devices and cloud servers, along with the response time for data to travel back and forth, is limited by network constraints and the speed of light, with additional delays caused by network congestion. These round-trip delays make a cloud-only model unsuitable for low-latency, real-time applications such as Industrial Automation, Augmented Reality (AR), Connected Vehicles, Internet of Things (IoT), and real-time online gaming, all of which require response times measured in milliseconds per user world.[1,2]

1.3. The Emergence of Fog Computing

Rather than replacing the cloud, fog computing acts as an augmentation. It is defined as a decentralized, virtual computing layer within the cloud, positioned at the network's edge near data collection and storage.[1] Fog computing provides computational, storage, and network services. This era, along with traditional clouds, represents more than mere technological progress; it brings about a fundamental shift in architecture driven by data transmission physics. As a networked paradigm, fog computing offers resources and services on a global scale. It transforms core geopolitical relationships concerning cloud technology into an integral part of data architecture.

2. The Hybrid Cloud-Fog Architecture

The integration of these two platforms yields a robust, hierarchical hybrid cloud-fog architecture, typically comprising three distinct layers.

- The IoT/Edge Layer: The Bottom layer is interfaced with the data-collecting sensors through the physical environment and the executors of actions and actuators.

- The Fog Layer: This middle layer consists of a distributed network of fog nodes. These nodes can be on any computing device, including edge servers, gateways, routers, or switches. This layer handles real-time filtering, data processing, and data aggregation.
- The Cloud Layer:The top layer includes traditional, centralized data centers with extensive computing and storage resources. The cloud manages resource-intensive analytics, complex data processing, and long-term data storage.

This multi-layered structure enables the systematic and productive management of processing activities. Time-sensitive activities are handled at the fog layer, while the cloud manages heavy, non-urgent processing activities.

3. The Importance of the Service Placement Problem (SPP)

The fog cloud architecture's dispersed and varied characteristics present a fundamental optimization challenge: identifying the optimal deployment location for services or modules of an Internet of Things (IoT) application. This is known as the Service Placement Problem (SPP).[5] The SPP is a complex combinatorial optimization problem, widely recognized as NP-hard, which indicates that identifying a provably optimal solution is computationally infeasible for any non-trivial system size.[5] This challenge is the primary reason why much effort has focused on developing reasonable heuristic and metaheuristic approaches. The placement decision is crucial because it significantly impacts system efficiency, which depends on factors such as application latency, network traffic, energy consumption, and the quality of service (QoS) for users.

3.1. Principal Issues about Service Placement

Deploying services in a hybrid cloud-fog architecture presents several important and interconnected challenges. This research will focus on these three challenges.

3.2. Challenge 1: Managing Resource Dissimilarity and Limitations

How can services be optimally mapped to a dynamic and varied subset of resource-constrained fog nodes while meeting application performance goals? [5]

A key characteristic of the fog layer is its noticeable diversity. Unlike the relatively uniform cloud data centers with resource pools that seem endless, the fog consists of a vast number of geographically spread devices with significantly different processing and storage capabilities. These fog nodes can include powerful micro-datacenters and edge servers, but they also include resource-limited devices such as network gateways, switches, and set-top boxes. Each of these nodes has limited CPU, memory, and storage.

The intricacy of the decision evolves due to diversity. An algorithm for the placement must undergo an exhaustive refinement of the criteria needed for each service in an application (for instance, CPU cycles, RAM that needs to be allocated to each fog node), fog nodes, and the limited, disparate resources in the nodes, and the mismatch. Speed placement could lead to degradation of service functionality or performance due to overloading a limited node, and in turn, grossly overestimating the underutilized capacity of the restricted nodes. A successful, intelligent, and well-reasoned placement will remain oriented toward the infrastructure’s key attributes, aiming for a viable and intelligently designed conjecture.

3.3. Challenge 2: "Achieving Service Continuity for End Users on the Move

What are the best strategies for dynamic placement and migration to reduce latency and service interruption in Mobile IoT applications?

There is a wide range of IoT devices supporting both mobile users and devices. Examples of such applications include connected vehicles, asset tracking, augmented reality (AR) on smartphones, and health wearables. The ability to provide static service is a limitation that restricts the full potential of these applications. The quality of service (QoS) decreases as the endpoint fog node hosting the application service for the mobile user moves. The increased distance results in higher latency and reduces overall capability. QoS ultimately constrains the application's timing requirements.

The theoretical solution to the problem is to migrate the fog computing service on the go to a node much closer to the user’s new location. However, service migration is a resource-intensive operation. It imposes overhead in terms of migration delay, increased network traffic, and energy consumption. One of the most important tasks is to develop intelligent migration policies that precisely define when and where migration is necessary. The most frustrating part of the process is avoiding “invalid migrations,”

which occur when a user moves out of the new area, expected to be the new zone of interest, before the migration process is complete, rendering the entire process pointless.

3.4. Challenge 3: Balancing Conflicting Quality of Service(QoS) Objectives

How can placement algorithms efficiently balance multiple competing objectives, such as latency, energy, and operational costs?

'Optimal' is a relative term whose meaning is rarely fixed. It is almost always a compromise among objectives. For example, minimizing latency involves placing a service on the closest fog node with the most processing power. However, this might be the most energy-inefficient node or one with the highest operational costs. Conversely, trying to minimize both costs and energy consumption might lead to choosing a node that is farther away, which can cause unacceptable delays. As a result, the placement is often suboptimal.

Different IoT applications have varying priorities. For instance, a real-time video analytics application for public safety must minimize latency. Conversely, an analytics application on a battery-powered environmental sensor node may be limited by energy constraints, making energy efficiency the primary design focus. Therefore, the challenge is to develop multi-objective optimization strategies. These strategies need to address the fundamental trade-offs between competing objectives simultaneously. Often, this results in multiple Pareto-optimal solutions, allowing a system operator to select the most suitable compromise among competing goals rather than a single, 'best' solution.

It is important to recognize these three challenges as interconnected aspects of the system. Each challenge tends to exacerbate the others. For example, a mobility user (Mobility) moving across a network may encounter fog nodes with varying capability levels (Heterogeneity). Service migration for a user must account for the resource limitations of the potential destination node, which alone makes it a Multi-Objective QoS problem. How does the system determine that the delay in data transmission is lower than the migration energy costs and the possibly less capable node that might be the target? As a result, the most advanced work in this field aims to formulate and address these challenges in an integrated way, making it highly sought after.

4. Research and Recommendations: A Review of Mitigation Strategies

To address the challenges outlined above, the research community has developed a wide range of algorithmic and policy solutions. Most strategies are explicitly tailored to the problem they aim to solve.

4.1. Strategies for Heterogeneous Environments

Given the NP-hard complexity of the SPP problem, exact optimization methods for large-scale, dynamic systems must be set aside. This practical necessity explains the rise of heuristics and metaheuristics, which are able, within reasonable time constraints, to access high-quality, near-optimal solutions.

4.1.1. Heuristic and Metaheuristic Approaches

- **Genetic Algorithms (GA):** Genetic algorithms (GA) are among the most popular methods in metaheuristics and belong to the class of population-based techniques inspired by natural selection. In the case of SPP, a potential placement solution (mapping services to fog nodes) is represented as a "chromosome." In GAs, the population of such solutions undergoes evolutionary processes over successive generations. These processes involve crossover and mutation strategies to populate schemes, helping to explore large solution spaces effectively. Each chromosome is evaluated by a fitness function based on a set of QoS parameters (latency, resource utilization, etc.). This evolution-driven selection process narrows the search area to focus on near-optimal placements that accommodate the resource heterogeneity of fog node constraints.
- **"Particle Swarm Optimization (PSO):** PSO is another nature-inspired metaheuristic that models the social behavior of a flock of birds. Each 'particle' represents a potential solution moving through a multi-dimensional Cartesian space. Every particle is influenced by its own best position and the best position of the entire swarm, affecting its velocity and direction. Accelerated PSO (APSO) is a variant of PSO that has been successfully used for multi-objective service placement and computation offloading problems. These works offer solutions that effectively balance delay, operational cost, and energy consumption, which are conflicting metrics.

4.1.2. ResourceAware Policies for Placement

- Edgeward Placement: This is the first and simplest heuristic policy based on the idea of placing application services geographically near the edge of a network (i.e., close to the end-users) to reduce latency. The algorithm starts from the lower tier of the network and, similar to a persistence search, gradually explores upward until it finds a host such as in a

local fog node with limited resources that has a tier suitable for the service's resource needs. Placement, while primarily used for simplicity, serves as a baseline for achieving lower latency. Although the strategy is straightforward, it is widely used for baseline latency reduction.

- Community Based Clustering: This technique addresses the lack of structure in a large, flat fog network. Geographically close or logically related fog nodes can be grouped into `communities' or `colonies'. The SPP is then simplified into a two-stage process, where the first stage assigns the entire application to the community where it can be best utilized. In the second stage, the individual application services are placed on nodes within that community's resource pool. This approach effectively reduces `lost space' and improves resource availability and fault tolerance through localized resource management utilization.

4.1.3. Formal Optimization Techniques

- *Integer Linear Programming (ILP):* For smaller-scale or static placement scenarios, the Service Placement Problem (SPP) can be formally modeled as an Integer Linear Programming (ILP) problem. Decision variables correspond with the distribution of services to specific nodes, while a set of linear equations and inequalities depict the described limitations of the resources and the required Quality of Service. In this model, the objective is to minimize or maximize a certain metric (for instance, the total latency). Although this approach can be tedious and less suitable for more dynamic and larger industries, ILP models remain influential in the optimization field, as they provide provably optimal solutions. Such models serve as the ideal benchmark for assessing the quality of heuristic and metaheuristic algorithms, as well as the overall process performance.
- The tactic selected in this scenario involves a deliberate approach that balances solution optimality with computational resource constraints. While Integer Linear Programming (ILP) provides the most optimal solution, the associated time complexity for real-time decision-making in extensive fog networks makes heuristic methods more practical. This contrasts with approaches that aim to guarantee optimality. It is generally agreed that the most promising strategy involves developing hybrid models that incorporate the advantages of multiple methodologies.

4.2. Mobility-Aware Service Placement and Migration

To ensure the preservation of voice quality and the overall user experience for mobile users, the strategy for service placement must be sufficiently adaptable and responsive to facilitate real-time adjustments to the service location via service migration.

4.2.1. Dynamic Migration Frameworks

The core principle of mobility management is to maintain proximity to the user by migrating the Virtual Machine (VM) or container hosting the service to a different fog node that is closer to the user. In the research, there is a distinction between reactive migration, which occurs after detecting user movement, and proactive migration, which aims to predict movement patterns and start the migration well before the user reaches the new location to ensure service quality [8,9].

4.2.2. Predictive Models for User Mobility

The key change in migration efficiency is the implementation of predictive models. A key concept is predicting sojourn time, which refers to estimating how long a user is likely to stay within reach of a newly connected access point.[8] The system can perform "intelligent" analysis by comparing this sojourn time to the predicted time for successful service migration. If the system determines that the predicted sojourn time is shorter than the migration delay, it considers the migration 'invalid" and will not proceed. The filters are crucial in preventing the system from wasting critical network and computational resources on so-called migration failure attempts, which, from the user perspective, provide no value. This enhances resource efficiency and improves overall system stability.

4.2.3. Case Study: Vehicular Fog Computing (VFC)

VFC is a complex area of mobility management that considers vehicle speed and unpredictable behavior. It remains at the forefront of mobility support, with strategies such as migration-aware interference (which accounts for local network contention from surrounding vehicles when selecting a migration target) and statistical model-based trajectory forecasting (using tools like Hidden Markov Models to predict a vehicle's path and plan for route servicing). [4]

The development of these strategies demonstrates a growing shift from simple, reactive approaches to intelligent, proactive ones. An effective management system considers the user's current location, their predicted path, real-time network conditions, and the availability of resources at potential target nodes to make optimized and refined migration decisions.

4.3. Multi-Objective Optimization for QoS Provisioning.

Effective service placement must balance multiple competing objectives simultaneously.

4.3.1. Pareto-Optimal Approaches

NSGA-II (Non-Dominated Sorting Genetic Algorithm II): To address other objectives, such as minimizing energy consumption and latency, a multi-objective optimization algorithm is employed. Within this frame, one of the most popular, powerful, and widely used genetic algorithm is NSGA-II. It does not provide only one "best" solution to a problem but a set of Pareto-optimal solutions. A solution is said to be Pareto-optimal if one of the objectives cannot be improved without degrading at least one of the other objectives. With this, system operators can select a placement policy that aligns with the system's operational priorities, given the trade-offs.

4.3.2.Policy-Based Prioritization

Most Delay-sensitive Application First (MDAF): This heuristic provides a practical and efficient solution for managing QoS in a deadline-driven environment. The algorithm's principle is straightforward: all incoming application deadlines are sorted in ascending order and stored in a list. The aim at this stage is to prioritize the least time-consuming among the "time-critical" applications by assigning them to the closest and most capable resources (e.g., local fog nodes and cloud servers as primary remote servers). For deadline-sensitive applications, this simple approach has been successfully used to prioritize deadlines over other objectives.

4.3.3."Mapping vs. Edge-ward Policies.

The more placement policies are compared, the more differences become the focus of optimization. The Edge-ward policy, described earlier, would offload tasks to the cloud when local fog resources are insufficient, whereas the Mapping policy would enqueue tasks on a fog node regardless of temporary overload. In some cases, the Edge-ward policy is easily outperformed by the Mapping policy. It takes a more cost-effective approach by accepting a local processing delay in a queue, thereby avoiding a higher latency penalty, since the cloud is a much more expensive resource. This demonstrates that the most suitable policy depends on the actual workload and network conditions.

All trends in QoS assistance appear to be moving toward more advanced, policy-driven systems. It is clear that relying on a single, static method to optimize this for the IoT world will not be enough. New placement systems are being developed that consider the specific needs of applications, such as placing policies that optimize fog networks to reduce latency for certain data streams or implementing more energy-efficient policies for battery-powered sensors. This principle of policy placement optimization, guided by real-time context, is a mature area of research and emphasizes the importance of increased context awareness.

5. Performance Analysis and Case Study Insights

The effectiveness and efficiency of different methods of service placement are best demonstrated and understood through quantitative results from various empirical studies. Given the scale and complexity of fog environments, it is understandable that simulation is the main method used for performance evaluation.

5.1. Common Evaluation Methodologies and Simulators

One commonly discussed simulation toolkit in the literature is iFogSim. [7] It extends the CloudSim toolkit and is specifically designed for modeling and simulating the hierarchical structure of fog and edge computing. It offers features to define the physical topology (such as cloud data centers, fog nodes, and sensors), network link parameters (including latency and bandwidth), and the framework of an IoT application as a set of integrated modules. Notably, this simulator enables users to create and test new policies for resource management and placement across different levels in a resource cloud, and to evaluate these policies using performance metrics like end-to-end latency, network energy and traffic, and total energy consumption. This capability makes it one of the most frequently cited simulation tools for testing and refining strategies in a consistent and controllable manner.

5.2. Quantitative improvements from case studies.

The intelligent service placement strategies, as illustrated and proven in several key simulation-based studies, are what distinguish them from standard service placement strategies. These studies demonstrate impressive performance improvements across various application domains.

Regarding mobile users, a 2019 study demonstrated the effect of a dynamic policy that filters out unnecessary service migrations by estimating a user's sojourn time to a new location. [8] This policy predicted a reduction in average latency of about

4ms, compared to static placement latency and 1ms over a simple "follow me" migration policy. The study clearly illustrated the impact of mobility management on latency.

In smart homes, where nearly every operational task is time-sensitive, a 2019 case study focused on a QoS policy called "Most Delay-sensitive Application First" (MDAF) [10]. Using the iFogSim simulator, the study showed that the MDAF policy could achieve selective deadline adherence better than the baseline Edge-ward and Cloud-only strategies. Additionally, the policy reduced cloud execution costs by over 62% compared to a cloud-only approach by better utilizing fog resources.

An analysis of e-Health systems in late 2019 compared three data placement algorithms and the "Mapping" algorithm, which delivered the best performance. When cloud resources were busy, queuing task sets at local fog nodes instead of instantly releasing them to the cloud resulted in the lowest execution times and cloud energy costs.

Lastly, the 2018 research in Vehicular Fog Computing developed a task allocation policy that, in its design, optimally balances both latency and loss in quality. The simulation results indicate it can cut the average service latency by 27% for real-time vehicular applications, which is significant in such a dynamic, fast-paced, and high-demand environment.

Table 1: Case Study Improvements

Domain	Placement Strategy Evaluated	Key Metrics Improved	Quantitative Improvement
Mobile Users	Mobility-Aware Migration (with sojourn time prediction)	Average Service Latency	Reduced average latency by ~4 ms vs. No Migration and 1 ms vs. naive migration.
Smart Home (Deadline-Critical)	MDAF (Most Delay-sensitive Application First)	Deadline Adherence, Execution Cost	Achieved 100% deadline adherence. Reduced cloud execution cost by 62% vs. cloud-only.
e-Health Systems	Mapping Algorithm	Execution Time, Cloud Energy	Consistently lowest execution times and least impact on cloud energy consumption.
Vehicular Fog Computing	Latency & Quality Optimized Task Allocation	Average Service Latency	Decreased average service latency by up to 27%.

6. Conclusion

This survey focused on the latest developments in deploying services for latency-sensitive IoT applications within hybrid cloud-fog architectures. The study highlighted fog computing as a crucial addition to the centralized cloud model in the era of IoT. The shift from static placement policies to dynamic, context-aware, and multi-layered strategies is quite impressive. Due to the NP-hard nature of the Service Placement Problem, heuristics and meta-heuristics, especially designed for complex multi-entity environments, such as Genetic Algorithms and Particle Swarm Optimization, have become very popular. In mobile systems, the most advanced approaches involve creating predictive migration policies that use sojourn time estimates to narrow down the set of potential migration destinations. These policies represent a step forward from older, reactive migration techniques and are considered to have lower overhead compared to other methods. The use of Pareto-based approaches with NSGA-II and priority setting models like MDAF has made it easier to optimize latency, energy, and cost simultaneously. The referenced simulation studies support the effectiveness of these strategies by clearly demonstrating lower latency, reduced costs, and improved deadline compliance compared to edge-only or cloud-only solutions.

References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in Proc. First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, Aug. 2012, pp. 13–16. [Online]. Available: https://doi.org/10.1145/2342509.2342513

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, Oct. 2016. [Online]. Available:(https://doi.org/10.1109/JIOT.2016.2579198)

[3] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," Journal of Systems Architecture, vol. 98, pp. 289–330, Sep. 2019. [Online]. Available: https://doi.org/10.1016/j.sysarc.2019.02.009

[4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017. [Online]. Available:(((((https://doi.org/10.1109/COMST.2017.2745201)))))

[5] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner, "Optimized IoT service placement in the fog," in Service-Oriented and Cloud Computing, Cham: Springer International Publishing, 2017, pp. 185–193. [Online]. Available:(http://www.infosys.tuwien.ac.at/Staff/sd/papers/ICFEC_2017_O_Skarlat.pdf)

[6] M. Taneja and A. Davy, "Resource aware placement of IoT application modules in Fog-Cloud computing," in 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, May 2017, pp. 1222–1228. [Online]. Available: https://dl.ifip.org/db/conf/im/im2017special/208.pdf

[7] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments," Software: Practice and Experience, vol. 47, no. 9, pp. 1275–1296, Sep. 2017. [Online]. Available: https://doi.org/10.1002/spe.2509

[8] S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer, and K. K. Leung, "A Mobility-Aware Dynamic Service Placement Policy for Edge Computing," in 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, Jul. 2019, pp. 1536–1546. [Online]. Available: https://pdfs.semanticscholar.org/0b09/9b675816046194a4e6a8078b4d8f96566a4c.pdf

[9] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and N. Parashar, "Mobility-aware application scheduling in fog computing," IEEE Cloud Computing, vol. 4, no. 2, pp. 26–35, Mar. 2017. [Online]. Available: https://doi.org/10.1109/MCC.2017.27

[10] A. M. A. El-Sayed, T. A. E. Salama, and M. A. Mohamed, "Deadline-Aware Failure Detection and Task Scheduling in IoT and Fog Computing," IEEE Access, vol. 7, pp. 145025-145034, 2019. doi: 10.1109/ACCESS.2019.2945281