Ad-hoc Cloud: A User-Provided Cloud Infrastructure at Network Edge

Huang Yaojun
yaojun.huang@aalto.fi

Tutor: Sara Ranjbaran

Abstract

As demands for processing complex tasks continue to rise and device capabilities continue to advance, resource sharing and collaboration among user devices become increasingly critical. In this scenario, traditional computing paradigms face challenges in effectively utilizing resources. Ad-hoc Cloud is a novel user-provided infrastructure concept that leverages existing and non-exclusive edge devices to achieve resource sharing. This paper provides an overview of Ad-hoc Cloud Computing, exploring various implementations, and highlighting their contributions to optimizing resource utilization and enhancing user experience. Additionally, the paper concludes several challenges and future research directions in Ad-hoc Cloud Computing, including user incentives, task delegation, and security. Despite being in its early stages, Ad-hoc Cloud Computing shows promise in supplementing existing computing paradigms.

KEYWORDS: User-Provided Cloud Infrastructure, Ad-hoc Cloud Computing, Edge Computing, Task offloading

1 Introduction

Although hardware technology continues to advance, individual machines still cannot keep up with the increasing demands of processing complex tasks. Cloud Computing (CC), a novel computing model [1], emerges as a solution to this challenge by revolutionizing the delivery of applications over the Internet. It offers virtually unlimited computing resources that can be provisioned on-demand, empowering users with unprecedented flexibility [2]. Mobile Cloud Computing (MCC) is a paradigm that combines Mobile Computing and Cloud Computing [3]. By outsourcing computing tasks to data centers, thin devices can deliver interactive experiences that exceed local capabilities. However, this model, while powerful, comes with a drawback: high latency.

Edge Computing and Fog Computing are models proposed to maximize the potential of Cloud Computing. These models aim to decentralize computing power and bring it closer to the user to reduce data transfer latency and enhance service responsiveness. Multi-Access Edge Computing (MEC) [4], seeks to improve application performance and user experience by processing relevant tasks in a place closer to users. However, compared to the centralized managed data centers, the decentralized management at the edge exacerbates the deployment challenges of edge servers. One of the challenges that prevent the feasibility of such a model is who will deploy such infrastructure.

The concept of User-Provided Infrastructure (UPI) has been developed to optimize system efficiency and utilize the potential of network edges. UPI makes use of idle heterogeneous edge devices that have increasing computing power to supplement edge computing capabilities. User devices cooperate to perform complex tasks beyond the capabilities of a single machine. Ad hoc Cloud Computing [5]. is one way to achieve UPI. It is a local cloud that utilizes computing resources from existing, non-exclusive, and unreliable infrastructures.

To evaluate the practicality of Ad-hoc Cloud Computing, this paper discusses the characteristics of Ad-hoc Cloud and reviews various Ad-hoc Cloud frameworks, attempting to summarize the challenges and directions in this area.

The paper is structured as follows. Section 2 provides an overview of the computing paradigms. Section 3 explores various research related to Ad hoc Cloud architecture. Section 4 discusses the approaches outlined in this paper. Finally, Section 6 concludes the paper.

2 Computing paradigm

Cloud computing is a model that offers access to a shared pool of computing and storage resources on-demand, ensuring scalability and accessibility from any location [2]. However, it faces significant challenges due to the geographical centralization of computing resources in distant data centers. This structure introduces transmission delays across the network, resulting in bottlenecks that are particularly challenging for latency-sensitive applications such as Augmented Reality and Cloud Gaming.

This section introduces Fog Computing and Edge Computing, two main computing paradigms that try to alleviate the problems of Cloud Computing and task offloading, the main approach to utilize the computing paradigm.

2.1 Fog Computing

Fog Computing (FC) was introduced by Cisco in 2012 as a response to the limitations of Cloud Computing [6]. It acts as an intermediary platform between conventional data centers and end devices, providing computing and storage services. Essentially, it represents a "descending" cloud that is closer to end-users.

Fog computing has several advantages over Cloud Computing, such as reduced bandwidth consumption, location awareness, low latency, and mobility support. The architecture of fog computing is typically geographically distributed and consists of a diverse array of heterogeneous devices with varying capabilities [7]. These unique properties enable the construction of large-scale automated systems, making fog computing a highly efficient solution for Internet of Things (IoT) implementations. FC has found widespread applications in areas such as smart cities.

2.2 Edge Computing

Edge Computing (EC) is another concept to extend the Cloud Computing paradigm. It is a distributed computing framework that leverages edge devices for data processing instead of offloading tasks solely to the cloud. By processing and analyzing data in place only one jump to its source, Edge Computing significantly reduces latency that occurs when data is transmitted to distant cloud centers, thus achieving shorter response times and enhanced user experience [8]. Edge Computing is not intended to replace cloud computing but rather acts as a supplement. By bringing computational capabilities to the network edge, it eliminates bottlenecks inherent in centralized architectures, thereby supporting the cloud computing experience. Multi-Access Edge Computing (MEC) [4] is an architecture defined by the European Telecommunications Standards Institute (ETSI). MEC provides cloud computing capabilities and IT services at the network edge, offering features such as on-premises deployment, lower latency, and location awareness [8]. The advantages of Edge Computing make it an essential solution for a variety of latency-sensitive applications, establishing its critical role in the fields of multimedia entertainment and smart homes.

2.3 Task offloading

Compared to data centers, end devices typically have limited computing power and may not be able to perform resource-intensive computing tasks locally, which prompts the adoption of task-offloading strategies. Task offloading involves delegating computing tasks to other nodes, aiming to reduce processing time, but potentially increase transmission time. There are three types of task offloading: local execution, partial offloading to other capable devices, and full remote computing [9]. The specific task offloading strategy chosen depends on various factors, such as the nature of the tasks, the characteristics of the processing devices, and optimization objectives like energy efficiency, latency, cost, security, and Quality of Experience [10].

Mobile Cloud Computing architectures involve processing tasks generated by end devices either locally or by sending them to cloud centers for processing. With advancements in hardware and the advent of 5G technology, end devices now have increased computing capabilities and reduced communication latency which allows tasks to be offloaded to the cloud or neighboring end devices for direct collaboration, known as Device-to-Device (D2D) Task Offloading. This approach increases flexibility and leads to a more diverse architecture. However, the joining of numerous heterogeneous devices distributed across different layers of the architecture makes the optimization problem intricate. Various methods have been proposed for optimized task scheduling, which can be categorized into three types: (a) Mathematical Optimization algorithms; (b) Machine learning algorithms; and (c) Control Theory-based approaches [11].

In the scenario of the Internet of Things (IoT), edge devices often exhibit different capabilities, so collaborative efforts are essential to efficiently execute tasks. For example, a camera may be tasked with image collection while a phone can be responsible for processing. By offloading tasks to other devices, end devices can work together to complete more complex tasks with shorter execution and transmission times.

3 Ad-hoc Cloud Computing

Ad hoc Cloud is an emerging research direction. It refers to a computational model that leverages existing, non-exclusive, and unreliable infrastructures to acquire computing resources [5]. Originating from the concepts of volunteer and grid computing, Ad hoc Cloud has the key following difference. It operates as a local cloud consisting of volunteer resources, eliminating the need for trust between users and infrastructure. Moreover, it ensures job continuity and minimizes interference with the host, adept at handling diverse workloads.

Ad hoc Cloud typically comprises a group of nearby mobile devices willing to share resources, alleviating the computational bottleneck at the edge server. Additionally, it provides the capability to execute compute-intensive applications locally when remote network connections encounter issues [12]. Ad hoc Cloud facilitates a cloud-style paradigm within the local network, effectively utilizing idle computing resources in the local network environment, thereby enhancing device utilization and reducing energy consumption [13].

This section presents a variety of implementations and applications of Ad hoc Cloud.

3.1 Dynamic Mobile Cloud Computing

Dynamic Mobile Cloud Computing [14] is a local cloud framework designed to optimize resource utilization in mobile computing. It includes three key components: resource handler, job handler, and cost handler. The resource handler is responsible for tasks such as resource discovery, monitoring, and metadata exchange. The job handler is tasked with managing jobs and scheduling while the cost handler estimates costs and handles transactions between client devices and the master device. In this framework, the master device initiates device discovery to identify potential clients. Subsequently, considering factors such as user priorities, requirements, and constraints, the framework calculates costs to select the optimal node for task delegation. Before or after job delegation, the master device pays the client device for its resources. The experiments use Bluetooth as the transmission solution and show that the offloading strategy can always provide better performance.

3.2 Transient Clouds

Transient Clouds [15] is a temporary network that forms dynamically among nearby devices, allowing them to act as a cloud platform and share their resources. A transient cloud is established when devices come together and dissipates when they leave. Resources shared in the Transient Cloud include general computing capabilities such as CPU and storage, and heterogeneous capabilities such as sensing and localization. Devices within the network submit tasks to the Transient Cloud, which then distributes the tasks based on the device's capabilities. The transient cloud employs a modified version of the Hungarian method, which dynamically adjusts the cost based on previous assignments to achieve load balancing. In the experimental simulation results using Wifi Direct and Android, Transient Clouds outperforms standalone execution when the ratio of computation to transmission size is high. This highlights the potential for collaborative resource sharing among users.

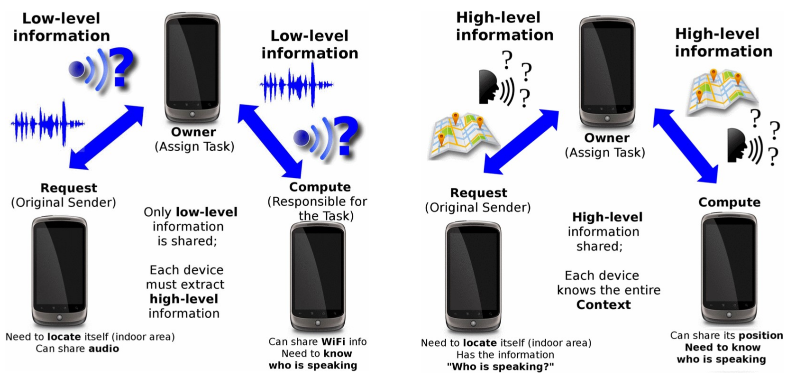

Figure 1. Comparison between traditional (left) and Context-Awareness (right) Transient Clouds [16]

Sciarrone et al. enhanced Transient Clouds by integrating the high-level information: context and introduced the concept of the Transient Context-Aware Cloud (TCAC) [16]. Within TCAC, devices share high-level information instead of low-level capabilities. This approach can significantly reduce the workload on the devices. For instance, if a device possesses knowledge of its position, it can share this information with other devices, thus obviating the continuous same task delegation to devices equipped with GPS functionality.

3.3 Ad-hoc Cloud Implementation

The paper [5] presents a comprehensive, integrated, and end-to-end Ad hoc Cloud solution, with an implementation based on BOINC. The overall architecture of the Ad-hoc Cloud is shown in Figure 2. It consists of cloudlets and ad hoc guests that utilize existing idle resources from host users to allocate them to tasks submitted by cloud users. An ad hoc client installed on the host manages the resource load and provides feedback on the host information to the server. This implementation follows the concept of Platform as a Service (PaaS), which ensures that the platform details are transparent to cloud users. The framework employs reliability as a metric for task scheduling, which is calculated by the previously assigned and completed jobs, host and guest failures, and resource load. To enhance system robustness, snapshots are periodically synchronized to potential backup machines, and in the event of a fault, one of the nodes is instructed to restore the snapshot for fault recovery.

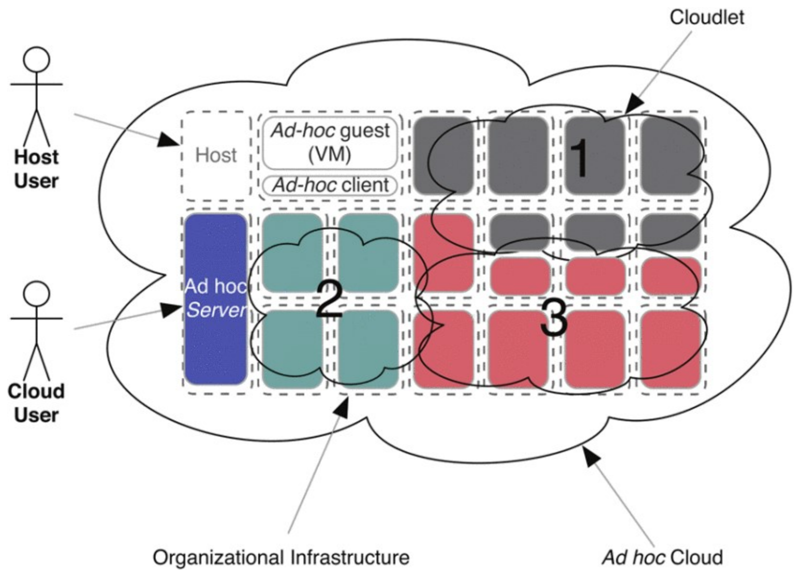

Figure 2. The Ad-hoc Cloud architecture [5]

Chi and Wang [17] extend the concept of Ad hoc Clouds into the domain of cloud gaming, introducing a progressive game resource download mechanism. In this framework, each end client possesses a copy of the initial game state and gradually downloads additional content from the cloud or nearby peers as users progress through the game. In addition, a collaborative task mechanism is employed to dynamically distribute game components across different devices. The ad-hoc cloudlet is leveraged for task execution initially, but when its capabilities are exceeded, tasks are transferred to the cloud. This approach optimizes resource utilization and effectively reduces energy consumption.

3.4 Crowdsourced Edge Computing

Crowdsourced Edge Computing (CEC) [18] extends the concept of Ad-hoc Cloud Computing to a larger scale at the network edge. CEC operates in an ad hoc manner, acting as a distributed computing framework where all peers are considered equal. Roles within the network are divided into three categories: Task handlers, Workers, and Message brokers, and each node takes on one or more of these roles. Compared to the prevailing commercial driving force of mainstream edge computing paradigms, the CEC model places a stronger emphasis on community and voluntary participation. In CEC, devices contribute computing resources based on the desires of individuals or communities. This vendor-neutral service fosters the creation of more sustainable communities and can lead to entirely new forms of digital economies. Such democratized resource sharing facilitates decentralization and holds significant potential for application in the IoT area. However, this appealing computing paradigm is still in its nascent stage of development and faces significant challenges in areas such as Quality of Service, data consistency and security, and resource management.

4 Discussion

Ad-hoc Cloud emerges as a promising frontier aimed at optimizing resource utilization and mitigating deployment challenges inherent in edge and fog computing. Despite various explorations in this domain, the paradigm is still in its infancy and awaits further investigations to reveal its full potential. The following are some research trends and directions:

1. User Incentives Mechanism: While conventional architectures rely on monetary incentives to drive users to join the network, recent research

has started to explore the integration of social computing. These novel approaches aim to encourage users to voluntarily contribute their devices, fostering a more sustainable ecosystem.

2. Task Delegation and Load Balancing: Task scheduling approaches have evolved from simple indiscriminate task offloading to account for the characteristics of heterogeneous devices, and further to a comprehensive task distribution that takes into account both device capabilities and load balancing. This evolution is crucial for the extension of Ad-hoc Cloud to the context of IoT environments, where collaboration among diverse devices is essential for efficient operation.

3. Scenario Analysis and Data Sharing: By analyzing tasks and data flow in practical applications, devices can be instructed to share contexts and individual information. This enables the Ad-hoc Cloud to make more optimal decisions at macroscopic scales, reducing redundant computations and data transmissions, which further optimizes resource utilization and reduces bandwidth consumption.

However, despite advances in areas such as network formation and task scheduling, Ad-hoc Cloud still faces significant challenges in terms of security and privacy. Ad-hoc Cloud does not assume the trustworthiness of devices within the network, allowing any node to join or leave the network at any time. This architectural flexibility makes it highly vulnerable to malicious attacks, including network contamination and data leakage. For Ad-hoc Cloud to be deployed effectively in the IoT domains such as smart cities, it is imperative to ensure the protection of participants' data. Therefore, further research in this direction is necessary to achieve a reliable and secure implementation of Ad-hoc Cloud.

5 Conclusion

This paper provides an overview of different implementations of the Ad-hoc cloud computing paradigm. Ad-hoc Cloud is a new model that uses idle resources to redefine edge computing infrastructures. The paper introduces two early-stage frameworks: Dynamic Mobile Cloud Computing and Transient Clouds, and presents a detailed implementation of Ad-hoc Cloud. It also discusses the concept of Crowdsourced Edge Computing.

In addition, the paper evaluates the current achievements of these approaches and identifies several future research trends, including the user incentive mechanism, task delegation, data sharing and security. Addressing these challenges has the potential to facilitate the utilization of Ad-hoc Cloud and transform the deployment paradigm of edge computing.

References

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy H Katz, Andrew Konwinski, Gunho Lee, David A Patterson, Ariel Rabkin, Ion Stoica, et al. Above the clouds: A berkeley view of cloud computing. Technical report, 2009.

[2] Peter Mell, Tim Grance, et al. The nist definition of cloud computing. 2011. doi: 10.6028/NIST.SP.800-145.

[3] Atta ur Rehman Khan, Mazliza Othman, Sajjad Ahmad Madani, and Samee Ullah Khan. A survey of mobile cloud computing application models. *IEEE Communications Surveys Tutorials*, 16(1):393–413, 2014. doi: 10.1109/SURV.2013.062613.00160.

[4] Sami Kekki, Walter Featherstone, Yonggang Fang, Pekka Kuure, Alice Li, Anurag Ranjan, Debashish Purkayastha, Feng Jiangping, Danny Frydman, Gianluca Verin, et al. Mec in 5g networks. *ETSI white paper*, 28(2018):1–28, 2018.

[5] Gary A. McGilvary, Adam Barker, and Malcolm Atkinson. Ad hoc cloud computing. In *2015 IEEE 8th International Conference on Cloud Computing*. IEEE, June 2015. doi: 10.1109/CLOUD.2015.153.

[6] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, page 13–16, New York, NY, USA, 2012. Association for Computing Machinery. doi: 10.1145/2342509.2342513.

[7] Firas Al-Doghman, Zenon Chaczko, Alina Rakhi Ajayan, and Ryszard Klempous. A review on fog computing technology. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 001525–001530, 2016. doi: 10.1109/SMC.2016.7844455.

[8] Vasco Pereira Jorge Bernardino Gonçalo Carvalho, Bruno Cabral. Edge computing: current trends, research challenges and future directions. *Computing*, 103:993–1023, 2021. doi: 10.1007/s00607-020-00896-5.

[9] Junna Zhang and Xiaoyan Zhao. An overview of user-oriented computation offloading in mobile edge computing. In *2020 IEEE World Congress on Services (SERVICES)*, pages 75–76, 2020. doi: 10.1109/SERVICES48979.2020.00029.

[10] S. Taheri-abed, A.M. Eftekhari Moghadam, and M.H. Rezvani. Machine learning-based computation offloading in edge and fog: a systematic review. *Cluster Computing*, 26:3113–3144, 2023. doi: 10.1007/s10586-023-04100-z.

[11] Firdose Saeik, Marios Avgeris, Dimitrios Spatharakis, Nina Santi, Dimitrios Dechouniotis, John Violos, Aris Leivadeas, Nikolaos Athanasopoulos, Nathalie Mitton, and Symeon Papavassiliou. Task offloading in edge and cloud computing: A survey on mathematical, artificial intelligence and control theory solutions. *Computer Networks*, 195:108177, 2021. doi: 10.1016/j.comnet.2021.108177.

[12] Ibrar Yaqoob, Ejaz Ahmed, Abdullah Gani, Salimah Mokhtar, Muhammad Imran, and Sghaier Guizani. Mobile ad hoc cloud: A survey. *Wireless Communications and Mobile Computing*, 16(16):2572–2589, 2016. doi: 10.1002/wcm.2709.

[13] Graham Kirby, Alan Dearle, Angus Macdonald, and Alvaro Fernandes. An approach to ad hoc cloud computing, 2010. doi: 10.48550/arXiv.1002.4738.

[14] Niroshinie Fernando, Seng W. Loke, and Wenny Rahayu. Dynamic mobile cloud computing: Ad hoc and opportunistic job sharing. In *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, pages 281–286, 2011. doi: 10.1109/UCC.2011.45.

[15] Terry Penner, Alison Johnson, Brandon Van Slyke, Mina Guirguis, and Qijun Gu. Transient clouds: Assignment and collaborative execution of tasks on mobile devices. In *2014 IEEE Global Communications Conference*, pages 2801–2806, 2014. doi: 10.1109/GLOCOM.2014.7037232.

[16] Andrea Sciarrone, Igor Bisio, Fabio Lavagetto, Terrence Penner, and Mina Guirguis. Context awareness over transient clouds. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–5, 2015. doi: 10.1109/GLOCOM.2015.7417785.

[17] Fangyuan Chi, Xiaofei Wang, Wei Cai, and Victor C.M. Leung. Ad hoc cloudlet based cooperative cloud gaming. In *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, pages 190–197, 2014. doi: 10.1109/CloudCom.2014.112.

[18] Stéphane Kündig, Constantinos Marios Angelopoulos, Sanmukh R. Kuppannagari, José Rolim, and Viktor K. Prasanna. Crowdsourced edge: A novel networking paradigm for the collaborative community. In *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 474–481, 2020. doi: 10.1109/DCOSS49796.2020.00080.