applied sciences

MDPI

Article

Scalable Dew Computing

Marjan Gusev

Faculty of Computer Science and Engineering, Ss Cyril and Methodius University, 1000 Skopje, North Macedonia; marjan.gushev@finki.ukim.mk

Abstract: Dew computing differs from the classical cloud and edge computing by bringing devices closer to the end-users and adding autonomous processing independent from the Internet, but it is still able to collaborate with other devices to exchange information on the Internet. The difference is expressed also on scalability, since edge and cloud providers can provide (almost endless) resources, and in the case of dew computing the scalability needs to be realized on the level of devices, instead of servers. In this paper, we introduce an approach to provide deviceless and thingless computing and ensure scalable dew computing. The deviceless approach allows functions to be executed on nearby devices found closer to the user, and the thingless approach goes even further, providing scalability on a low-level infrastructure that consists of multiple things, such as IoT devices. These approaches introduce the distribution of computing to other smart devices or things on a lower architectural level. Such an approach enhances the existing dew computing architectural model as a sophisticated platform for future generation IoT systems.

Keywords: edge computing; dew computing; serverless computing; function as a service; device as a service; thing as a service

check for updates

Citation: Gusev, M. Scalable Dew Computing. *Appl. Sci.* **2022**, *12*, 9510. https://doi.org/10.3390/app12199510

Academic Editor: Juan Francisco De Paz Santana

Received: 24 July 2022
Accepted: 2 September 2022
Published: 22 September 2022

1. Introduction

Classical cloud computing [1,2] focuses on providing servers to process complex computations, and does not include Internet of Things (IoT) devices [3,4]. Post-cloud systems [5,6] upgrade the classical cloud computing systems to bring the computation closer to the user, locating smaller servers nearby the user such asdata consumers at the network edge [7]. These architecture models appear in different forms, such as cloudlets [8], fog computing [9], edge computing [10], mobile edge computing [11], and dew computing [12].

Edge computing [13] improves the response times and saves bandwidth in a distributed computing platform, while the dew computing [12,14,15] extends it by adding independence and collaboration features and providing the autonomous processing [16], where instead of being on the edge, the devices can perform autonomously out of the edge.

Serverless computing is the approach when the users are relieved to take care of the resources, transferring the problem to the cloud providers. Instead of requesting the resources, the end-users are just specifying the functions to be executed, and the cloud provider ensures relevant resources. It simplifies the way the users look at the cloud computing [17,18] by enabling a platform [19] where the user does not need to request more or fewer servers, transferring this issue to the provider. Since dew computing targets autonomous processing on the device itself, these architectures are not able to provide scalability.

The deviceless computing approach [20], also recognized as a serverless approach for IoT [21], was applied for streaming IoT applications [22] focusing on the "traditional" edge server solution, and in this paper, the emphasis is on the dew computing and providing the scalability.

The rest of the paper has the following structure. Related work is elaborated in Section 2. Section 3 presents the architecture model, analyzing it for dew computing systems, and Section 4 presents the new implementations of scalability and hardwareless computing. A use case and analysis of challenges and limitations are discussed in Section 5. Finally, Section 6 presents relevant conclusions and directions for future work.

2. Related Work

Scalability is the ability of an analyzed system to handle an increasing amount of workload, both in terms of processing power and storage resources, exploiting its potential to be easily expanded to accommodate the growth of demands [23], or tendency to handle various computing resources, including applications and communication infrastructure [15]. Scalable distributed computing systems evolved through the cloud, then later fog, towards dew computing [14]. To be scalable, dew computing must penetrate into sundry networks, vigorous end-user machines, and vibrant dew-system assemblies and disseminate a prominent sensor-aware performance in case of sudden growth in the dew end-users and applications [24].

Scalability of dew computing was expected to be a promising feature without going into depth analysis [25], extracting that extreme scalability is offered in dew computing by processing distributed in multiple local devices in the user's proximity. Dew computing systems can switch between upgrading and downgrading resources according to users' needs, such as switching between multi-core and single-core processors [26]. Scalability and elasticity at the level of edge and dew computing can be provided by realizing a negotiation protocol and assigning tasks to other smart devices on the edge of the network, or even going deeper to the end-user devices [27].

A model of a dew computing solution for IoT devices was developed, addressing vertical scalability to higher-level cloud-based servers [28], to minimize data latency and mitigate the data processing burden on a centralized server. Performance was evaluated on developed real-time processing IIoT communication services applied on a distributed and scalable computing hierarchy [29], also mainly addressing vertical scalability.

The DeWatch architecture and framework were introduced to integrate smartwatches on a dew computing level and achieve a more scalable system [23], performing direct communication with nearby dew devices (from desktops, laptops, smartphones, smart home devices, and even with IoT smart sensors and devices) that are in proximity of the smartwatch ported by the user. This platform addresses both the vertical and horizontal scalability aspects and analyzes only the benefits of cloud computing service decentralization, with impacts on Big Data IoT processing.

A formal definition of "less" concepts is introduced in an earlier paper [30], where "less" means that the user is relieved of the need to worry about computing resources, which will be provided by a nearby server, smart device, or an IoT embedded system. Therefore, the serverless concept corresponds to servers in a cloud, and the deviceless corresponds to nearby devices in an IoT environment. In the serverless scenario, the user writes functions and the cloud provider ensures server resources to execute them. In the deviceless scenario, the user writes functions and the provider of the nearby IoT infrastructure ensures device resources to execute them.

The serverless platform provider accepts the burden to scale the software as needed in an elastic way to handle any kind of unanticipated load [31], while the users do not worry controlling the way their software is administered. A serverless real-time data analytics platform [32] is recommended for horizontal architectural designs between edge devices, especially for streaming data and integration of IoT.

Available serverless implementations include, at least, Amazon Web Services (AWS) Lambda, Azure Functions, and Google Cloud Functions (GCP). Insights into architectures, resource utilization, and the performance evaluation of the scalability show a number of issues and limitations besides the declaration of unlimited elasticity and resource provision of serverless functions [33]. Some even claim that serverless computing moves one step forward and two steps back in a distributed computing environment [34].

Containers enable stateless functions to be started on request, providing automatic scaling and, in addition, automated suspension if they are no longer needed [35]. This allows rapid application deployment in dynamic and possibly heterogeneous edge and fog environments [36].

The deviceless approach fosters development of IoT applications to extend the serverless paradigm towards the network edge, conceiving data pipelines under a flow-based development environment leveraging a geographically distributed IoT infrastructure [21]. A model to improve the healthcare service is introduced, addressing the deviceless approach as an emerging paradigm and technology leveraging next-generation computing systems [37].

Independency of devices is analyzed from architectural concepts and requirements as a prerequisite to achieving elastically scalable deviceless systems [27]. Challenges, design aspects, and models of deviceless edge computing are addressed by implementing the serverless paradigm at the edge [38].

The state of the art for scalable IoT data processing was analyzed in our earlier paper [27], including a system that scales DSP across multiple cloud regions to enable low-latency geo-distributed data analytics [39], QoS-aware deployment of DSP operators onto decentralized resources [40], or osmotic flow model for IoT [41]. Function as a service (FaaS) approaches were also addressed for devices and servers at the edge [32,42,43].

3. Architecture Model

A simplified modern architectural model, presented in Figure 1, consists of cloud servers on the top layer, smart devices in the middle layers presenting the edge and dew computing servers, and "things", representing the IoT devices at the bottom layer [30].

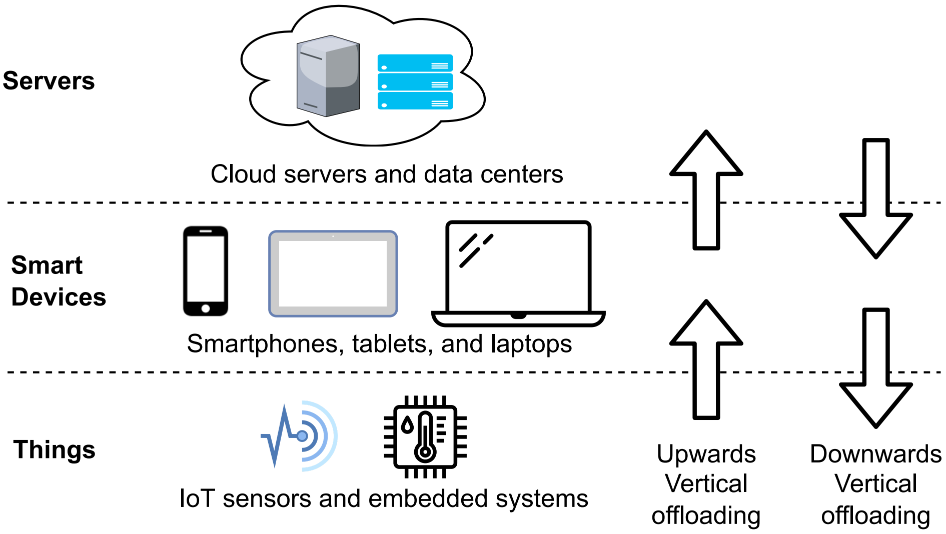
Servers
Cloud servers and data centers
Smart Devices
Smartphones, tablets, and laptops
Things
IoT sensors and embedded systems
Upwards Vertical offloading
Downwards Vertical offloading

Figure 1. A three-layer architecture model [30], which consists of servers, smart devices, and things.

Let us emphasize the difference of the possibility to exchange data between different layers and between different devices in the same layer. Smart devices are found at the edge of the network, which means they are always connected to the Internet, while the "things" work out of the edge and may also collaborate with smart devices and higher-level servers, fitting into the classical definition of dew computing.

The main difference between the edge and dew computing is the operation mode [16]:

- *Edge computing* includes smart devices found at the edge, which can always collaborate with higher-level servers (always connected to the Internet).
- *Dew computing* assumes that, in addition to the availability to communicate to the higher architecture levels, the smart devices and "things" (IoT devices) can work autonomously without communication to higher architecture levels.

This distinction between Internet connection availability and autonomous feature impacts the implementation of the scalability feature, which is analyzed next.

4. Results

4.1. Scalability

We define scalability as the ability of an analyzed system to provide sufficient performance even if there is an increase in the workload. Usually, a system scales by multiplying (replicating) the resources or by increasing the performance potential of the existing resources. This defines two basic approaches to achieving scalability:

- *Vertical scalability* is realized in an upwards vertical direction, usually by offloading computing requirements to more powerful computing resources in the computing architecture.
- *Horizontal scalability* is achieved by offloading computing requirements to more nearby computing resources.

In addition, we can also define a mixed approach as *diagonal scalability*, where both the vertical and horizontal scalability approaches are applied.

When these definitions are applied to the edge and dew computing architecture model presented in Figure 1, it is expected that the vertical scalability will be realized in the upwards direction offloading computations to the higher levels, where the resources are more powerful.

According to the offloading capabilities, our earlier paper [27] addresses two horizontal offloading methods, which are the basis to define two new possible horizontal scalability implementations for edge and dew computing:

- *Distributed horizontal scalability* solution (Figure 2), where offloading for scalability purposes is realized on nearby devices on the same architectural level.
- *Centralized horizontal scalability* solution (Figure 3), where a master device coordinates the offloading among devices on the same architecture level.

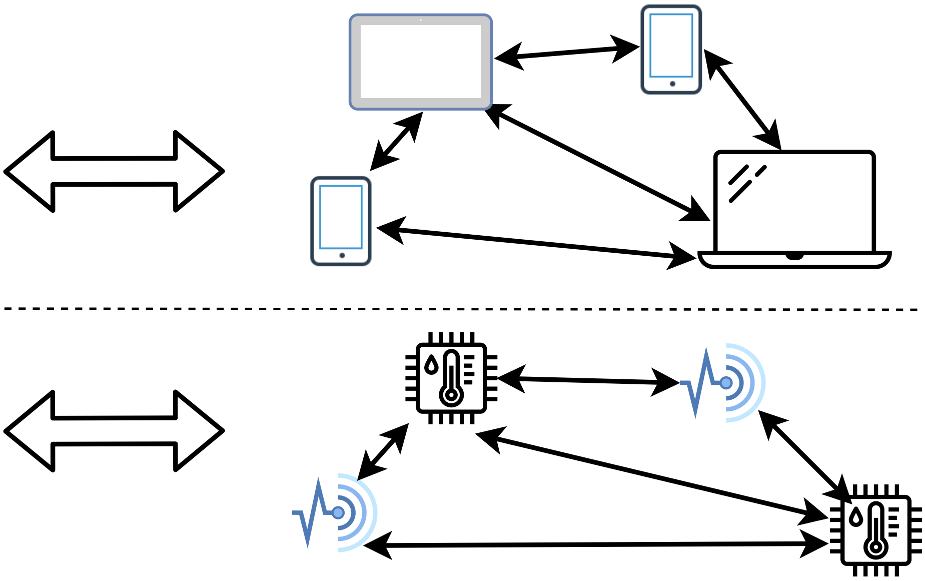

Figure 2. Horizontal scalability with distributed approach achieved by offloading to other devices on the same architectural level.

Distributed horizontal scalability is usually realized on the principles of ad hoc networks. Each device tries to connect to the other device on the same architecture level. In case of a centralized horizontal approach, all devices on the same architectural level share their resource availability to the master device, which can schedule tasks and orchestrate processing to the available devices [27]. The offloading procedure starts by checking if the neighboring device has sufficient energy resources and processing capacity to offload specific tasks, and then transfers the data, starts the computation, and retrieves back the results.

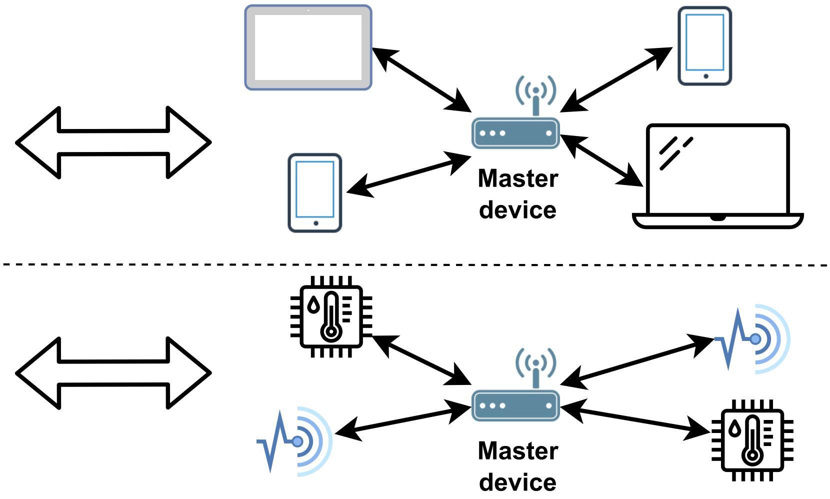

Figure 3. Horizontal scalability with centralized approach achieved by offloading to a master device on the same architectural level.

Horizontal scalability approaches can be realized on horizontal offloading schemes, such as those for cloud–edge systems analyzed by Ristov et al. [15] developing a horizontal scalable balancer, or those by Thai et al. [44], mainly from the aspect of workload and capacity optimization, focusing on service availability and capacity, workload, delay, and cost of applied system, or the approach elaborated by Deb et al. [45] as a two-step task distribution procedure. Flores et al. [46] expanded the traditional cloud offloading model to include an autoscaler component that acts as a load balancer for different computational resources and provides a horizontal offloading scheme. Several case studies include horizontal offloading mechanisms for IoT and post-cloud architectures, such as a traffic management system [47] and vehicle-to-vehicle networks [48]. Two models are proposed: vertical default and vertical (VDV) model and vertical default and horizontal shortest (VDHS) model, as a combination of vertical and horizontal offloading in the mobile edge environment [49].

The new vertical scalability approach for dew computing means to offload computations on a lower architecture level, as presented in Figure 1. In this case, computations are offloaded to a larger number of devices with smaller computation power.

Analyzing the architecture model in Figure 1, we conclude that vertical scalability is possible only in case the smart devices and "things" (IoT devices) are connected to the Internet. To support an autonomous performance, the smart devices and "things" (IoT devices) need to communicate to other devices at the same architectural level, meaning that the only alternative is horizontal scalability.

Classification of dew scalability was addressed [24] by (1) allowing dewlets among nondew systems; (2) defining virtual dew clusters (geographic expansions); and (3) utilizing cloud or fog or edge interplay. The first two options address horizontal scalability, while the third addresses the vertical. Option 1 addresses the distributed horizontal scalability, and option 2 the centralized horizontal scalability. There is no specification of a downwards vertical scalability.

4.2. Hardwareless Computing

Scalability is the essence of the "infrastructureless" or "hardwareless" concept, which is to relieve the user from taking care of required hardware resources and focusing on the application. The provider of computing resources is the one that knows better to manage the resources and can scale the system much faster to cope with the demand.

The *serverless approach* addresses the use of server instances, such as virtual machines, containers, or storage. The user just specifies the application to be a function that will be executed as a service. This concept is applicable for edge computing in the case when the "things" (IoT devices) communicate with smart devices, or when the smart devices offload computing requests over the Internet to the edge servers, which is realization of the vertical scalability.

The *deviceless approach* applies the same "less" idea on devices instead of servers, requiring specific functions to be executed on a nearby device, resulting in a seamlessly integrated application execution infrastructure of IoT and smart devices. In this sense, it is an application of horizontal scalability, achieved by offloading functions to nearby smart devices in the same architecture level. Such a concept is also known as **Device as a Service** [30], since a specific device will execute a required function. Examples of smart devices include, at least, those installed in cars, and can be used while the car is in a parking lot using solar energy.

The *thingless approach* is a new approach providing a **Thing as a Service** [30], in the case that "things" as IoT devices may also be powerful enough to execute functions. The idea is to engage "things" to execute functions instead of smart devices or servers. It is a kind of vertical scalability but in the opposite (downwards) direction, instead of upwards.

All these concepts implement scalable dew computing, especially when they work in the operational mode of an autonomous system without an Internet connection. For example, a dew device may find low battery power and would initiate a request to a nearby device to take care of executing functions. Finally, all these new concepts specify the real scenario of distributed computing resources in an ad hoc computing grid as a new computing trend.

5. Discussion

5.1. Use Case

Let us analyze an example of a heart monitoring system based on a wearable ECG sensor that streams data at a rate of up to 1 KB/sec, sending it to the cloud server for further processing and information sharing. The motivation problem is to realize simultaneous real-time heart monitoring by wearable ECG sensors for thousands of patients.

The solution is a typical dew computing solution, meaning that it involves all three layers of a "thing" (wearable ECG sensor), a smart device, such as an Android or iOS device, and a server application in a cloud. Besides the normal operation mode of communicating over the Internet for offloading data and computations to servers, the system may work totally autonomously and provide relevant information to the user, and exchange data whenever there is an Internet connection to provide information sharing to caregivers.

However, a single-user system simply may not replicate and act as a distributed multi-user as one might think. If a hospital would like to integrate a system that monitors thousands of patients, we need to ensure a scalable solution.

There are several issues concerning scalability:

- *Q1: Is it feasible?*
- *Q2: Is the resource consumption linear to workload requirements?*
- *Q3: What are the limits?*

The question Q1 means to find whether this scalability type is possible. If it is possible, how will it be achieved? Then, we find out if the growth of resources is proportional to the increase of demands, expecting it to be at most linear (answering Q2). The last question, Q3, addresses the limitations of such a system. We should analyze the problem not to just increase the hardware resources, but to organize the software to be able to cope with the increased demand.

Another use-case example of vertical downwards scalability can be a future implementation of a heterogenous computing resource sharing system based on a set of cars equipped with computing devices which can be exploited while they are being charged in a parking lot [30], or maybe to include smartphones while they are idle and being charged overnight.

Typical use-case examples of distributed horizontal scalability are based on offloading computations to neighboring smartphones on edge level [46], within a traffic management system [47], or between vehicles [48].

The main objective of this paper is finding out how to achieve scalability. This is a typical Big Data problem when data are coming from multiple sources in large volumes and velocities. Since our scope is not to provide a communication platform capable of accepting and collecting data from thousands of data streams, we focus on the computing infrastructure. If a single medium-sized container or virtual station can process tens of data streams simultaneously, we still need to find a solution that will scale hundreds of such computing resources.

The following scenarios are possible for vertical scalability to allow computation offloading and information sharing:

- High-performance data analytics (HPDA) platform with dedicated computing resources, activated according to the demand, thus implementing very powerful data center resources that can accept extremely large computing requests.
- Serverless solution using a cloud provider that takes care of cloud instances.
- Cloud solution with a specific workflow manager to take care of requirements.

The HPDA option can be optimized to cope with high processing demands but it cannot scale automatically, and it is not cost-efficient. The serverless solution and the cloud solution with dedicated workflow manager are more cost-efficient but are limited in providing computing resources. In the case of an approximately permanent number of data streams, the HPDA option is recommended, while the other two options are more convenient for a fluctuating number of data streams.

Besides these approaches, where the processing is realized on powerful servers in an HPC data center or a data center in a cloud, we can also use the edge/dew computing approach and distribute the processing closer to the user, so the cloud will be used only for information-sharing purposes. Horizontal scalability refers to autonomous dew computing operation mode and can be achieved independently by replication of the end-user systems. The deviceless and thingless approaches are recommended when the smart device detects low energy resources, so the processing can be offloaded to other devices and things, or when a single device is used to monitor several sensors.

The concept of downwards vertical computing is to provide a large number of smaller processing units to complete a task (function), which is an idea of processing via graphical processing units (GPUs) instead of central processing units (CPUs). Note that all applications will benefit from this approach, similar to the GPUs case.

5.2. Challenges

As the technology advances, soon, the “things” (IoT devices) may become smart and more powerful to be capable for both distributed and centralized horizontal scalability and downwards vertical scalability. To allow such functionalities, the dew computing devices need implementation of concepts of *device abstraction* and *server independence* [30], and to provide *collaboration* and *delegation* features with offloading of functions to other devices or things.

Advantages that the deviceless approach [30] brings to scalability include, at least:

- Fault tolerance;
- Availability to perform longer;
- Availability to be independent of the limited power supply.

A set of challenges to be solved by scalable dew systems [24] include the following: (1) the impartial functioning of dew computers requires preassessment because of the inadequate means of dew devices; (2) a dew system should ensure severance of resources in case it acts as an autonomous minimal cloud, as would happen; (3) intelligent controlling of the scalability opportunities for diverse consequences is perilous to the functioning of dew computers.

5.3. Limitations

Scalability is limited because it is dependent on the availability of computing resources. Although, today, cloud providers promote endless scalability opportunities by serverless computing, to integrate a function as a service solution, one needs to specify the hardware requirements of a cloud instance to process the function, and the cloud provider specifically limits the number of functions that can process the requirements simultaneously. For example, AWS and GCP limit to thousands of such computing instances.

Limitations are also present in the deviceless and thingless approaches. The number of nearby smart devices or things may be limited since it depends on their availability close to the user. Even in the best ubiquitous and pervasive scenario [50], we need to wait for more advanced technology solutions and provide many more IoT devices throughout the living environments. In addition to the availability of devices capable to offload and accept functions for execution, we face other legal aspects, such as making a smart contract and charging for the provided services. When cloud computing became available, a lot of providers started to provide resources for others, investing in large data centers. The next step is to invest in providing a scalable ubiquitous and pervasive environment that consists of multiple smart devices located in an environment with millions of dew devices around us.

6. Conclusions

We specified how the scalability can be implemented on a dew computing level. The usual method is to use the upwards vertical scalability when the user offloads computation and storage requirements to higher-architecture-level servers. Serverless computing is a modern solution when the cloud provider takes care of managing the computing resources, while the user specifies the functions to be processed.

In this paper, we introduced two types of horizontal scalability (distributed and centralized) and also the downwards vertical scalability. Horizontal scalability can be achieved by a distributed approach when the functions from a specific smart device are delegated to other smart devices on the same architecture level. Centralized horizontal scalability is achieved when there is a master node that controls the ad hoc network of smart devices, and conducts the orchestration of the processing requirements. In this sense, the deviceless approach is just an application of the serverless idea on the level of smart devices. The thingless approach is a new approach where the same idea is applied to the lowest architecture level among the "things" (IoT devices).

A new approach to achieving dew computing scalability is the downwards vertical scalability when the processing of functions is delegated to lower architecture levels, which may happen in the case of availability of a large number of smaller IoT devices.

Our research on scalability using the "hardwareless" approach by specifying deviceless and thingless approaches means establishing a platform where the providers manage the resources and users think only about using them. This is similar to the provision of utilities, such as the example of the airport waiting rooms that offer a lot of electrical outlets where the users can recharge their smartphones, or WiFi connection to access the Internet. They do not think about availability, they think only about using applications.

Future work will target more details in specifying the hardwareless computing requirements and implementation.

Funding: This research was funded by the European High-Performance Computing Joint Undertaking (JU) through the FF4EuroHPC project under grant agreement No 951745 for the realization of the experiment "CardioHPC-Improving DL-based Arrhythmia Classification Algorithm and Simulation of Real-Time Heart Monitoring of Thousands of Patients". The JU receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Italy, Slovenia, France, and Spain.

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. Fox, A.; Griffith, R.; Joseph, A.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I.; Zaharia, M. Above the clouds: A berkeley view of cloud computing. *Dept. Electr. Eng. Comput. Sci. Univ. Calif. Berkeley Rep. UCB/EECS* **2009**, *28*, 2009.
2. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I.; et al. A view of cloud computing. *Commun. ACM* **2010**, *53*, 50–58. [CrossRef]
3. Ashton, K. That 'internet of things' thing. *RFID J.* **2009**, *22*, 97–114.
4. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
5. Zhou, Y.; Zhang, D.; Xiong, N. Post-cloud computing paradigms: A survey and comparison. *Tsinghua Sci. Technol.* **2017**, *22*, 714–732. [CrossRef]
6. Pan, Y.; Thulasiraman, P.; Wang, Y. Overview of Cloudlet, Fog Computing, Edge Computing, and Dew Computing. In Proceedings of the 3rd International Workshop on Dew Computing, Toronto, ON, Canada, 29–30 October 2018; pp. 20–23.
7. Shi, W.; Dustdar, S. The promise of edge computing. *Computer* **2016**, *49*, 78–81. [CrossRef]
8. Satyanarayanan, M.; Bahl, P.; Caceres, R.; Davies, N. The case for VM-based cloudlets in mobile computing. *Pervasive Comput. IEEE* **2009**, *8*, 14–23. [CrossRef]
9. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and its role in the Internet of things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, 17 August 2012; ACM: New York, NY, USA, 2012; pp. 13–16.
10. Garcia Lopez, P.; Montresor, A.; Epema, D.; Datta, A.; Higashino, T.; Iamnitchi, A.; Barcellos, M.; Felber, P.; Riviere, E. Edge-centric computing: Vision and challenges. *ACM SIGCOMM Comput. Commun. Rev.* **2015**, *45*, 37–42. [CrossRef]
11. Patel, M.; Naughton, B.; Chan, C.; Sprecher, N.; Abeta, S.; Neal, A. *Mobile-Edge Computing Introductory Technical White Paper*; Mobile-Edge Computing (MEC) Industry Initiative; ETSI: Sophia Antipolis, France, 2014; pp. 1–36.
12. Wang, Y. Cloud-dew architecture. *Int. J. Cloud Comput.* **2015**, *4*, 199–210. [CrossRef]
13. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
14. Skala, K.; Davidovic, D.; Afgan, E.; Sovic, I.; Sojat, Z. Scalable distributed computing hierarchy: Cloud, fog and dew computing. *Open J. Cloud Comput.* **2015**, *2*, 16–24.
15. Ristov, S.; Cvetkov, K.; Gusev, M. Implementation of a Horizontal Scalable Balancer for Dew Computing Services. *Scalable Comput. Pract. Exp.* **2016**, *17*, 79–90. [CrossRef]
16. Gusev, M. What makes Dew computing more than Edge computing for Internet of Things. In Proceedings of the 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 12–16 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1795–1800.
17. Jonas, E.; Schleier-Smith, J.; Sreekanti, V.; Tsai, C.C.; Khandelwal, A.; Pu, Q.; Shankar, V.; Carreira, J.; Krauth, K.; Yadwadkar, N.; et al. Cloud programming simplified: A berkeley view on serverless computing. *arXiv* **2019**, arXiv:1902.03383.
18. Van Eyk, E.; Toader, L.; Talluri, S.; Versluis, L.; Uță, A.; Iosup, A. Serverless is more: From paas to present cloud computing. *IEEE Internet Comput.* **2018**, *22*, 8–17. [CrossRef]
19. Gusev, M. Edge and Dew Computing for Streaming IoT. In Proceedings of the 3rd International Workshop on Dew Computing, Toronto, ON, Canada, 29–30 October 2018; pp. 1–7.
20. Glikson, A.; Nastic, S.; Dustdar, S. Deviceless edge computing: Extending serverless computing to the edge of the network. In Proceedings of the 10th ACM International Systems and Storage Conference, Haifa, Israel, 22–24 May 2017; p. 1.
21. Benomar, Z.; Longo, F.; Merlino, G.; Puliafito, A. Deviceless: A Serverless Approach for the Internet of Things. In Proceedings of the 2021 ITU Kaleidoscope: Connecting Physical and Virtual Worlds (ITU K), Geneva, Switzerland, 6–10 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8.
22. Gusev, M.; Dustdar, S. Going Back to the Roots: The Evolution of Edge Computing, An IoT Perspective. *IEEE Internet Comput.* **2018**, *22*, 5–15. [CrossRef]
23. Garrocho, C.T.B.; Oliveira, R.A.R. Counting time in drops: Views on the role and importance of smartwatches in dew computing. *Wirel. Netw.* **2020**, *26*, 3139–3157. [CrossRef]
24. Ray, P.P. Minimizing dependency on internetwork: Is dew computing a solution? *Trans. Emerg. Telecommun. Technol.* **2019**, *30*, e3496. [CrossRef]
25. Wang, Y. Definition and Categorization of Dew Computing. *Open J. Cloud Comput.* **2016**, *3*, 1–7.
26. Ray, P.P. An introduction to dew computing: Definition, concept and implications. *IEEE Access* **2017**, *6*, 723–737. [CrossRef]
27. Gusev, M.; Koteska, B.; Kostoska, M.; Jakimovski, B.; Dustdar, S.; Scekic, O.; Rausch, T.; Nastic, S.; Ristov, S.; Fahringer, T. A deviceless edge computing approach for streaming IoT applications. *IEEE Internet Comput.* **2019**, *23*, 37–45. [CrossRef]
28. Seif, A.A.; El-Saber, N. Scalable distributed-computing iot applied architecture with semantic interoperable gateway. In Proceedings of the 3rd Africa and Middle East Conference on Software Engineering, Cairo, Egypt, 12–13 December 2017; pp. 43–44.

29. Garrocho, C.T.B.; da Cunha Cavalcanti, C.F.M.; Oliveira, R.A.R. Performance evaluation of industrial internet of things services in devices of cloud-fog-dew-things computing. In Proceedings of the 2020 X Brazilian Symposium on Computing Systems Engineering (SBESC), Florianópolis, Brazil, 24–27 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8.
30. Gusev, M. Serverless and Deviceless Dew Computing: Founding an Infrastructureless Computing. In Proceedings of the 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 12–16 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1814–1818.
31. Kritikos, K.; Skrzypek, P. A review of serverless frameworks. In Proceedings of the 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), Zurich, Switzerland, 17–20 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 161–168.
32. Nastic, S.; Rausch, T.; Scekic, O.; Dustdar, S.; Gusev, M.; Koteska, B.; Kostoska, M.; Jakimovski, B.; Ristov, S.; Prodan, R. A Serverless Real-Time Data Analytics Platform for Edge Computing. *IEEE Internet Comput.* **2017**, *21*, 64–71. [CrossRef]
33. Wang, L.; Li, M.; Zhang, Y.; Ristenpart, T.; Swift, M. Peeking behind the curtains of serverless platforms. In Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC 18), Boston, MA, USA, 11–13 July 2018; pp. 133–146.
34. Hellerstein, J.M.; Faleiro, J.; Gonzalez, J.E.; Schleier-Smith, J.; Sreekanti, V.; Tumanov, A.; Wu, C. Serverless computing: One step forward, two steps back. *arXiv* **2018**, arXiv:1812.03651.
35. Baldini, I.; Castro, P.; Chang, K.; Cheng, P.; Fink, S.; Ishakian, V.; Mitchell, N.; Muthusamy, V.; Rabbah, R.; Slominski, A.; et al. Serverless computing: Current trends and open problems. In *Research Advances in Cloud Computing*; Springer: Berlin, Germany, 2017; pp. 1–20.
36. Becker, S.; Schmidt, F.; Kao, O. EdgePier: P2P-based Container Image Distribution in Edge Computing Environments. In Proceedings of the 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC), Austin, TX, USA, 28–30 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8.
37. Tuli, S.; Tuli, S.; Wander, G.; Wander, P.; Gill, S.S.; Dustdar, S.; Sakellariou, R.; Rana, O. Next generation technologies for smart healthcare: Challenges, vision, model, trends and future directions. *Internet Technol. Lett.* **2020**, *3*, e145. [CrossRef]
38. Nastic, S.; Dustdar, S. Towards deviceless edge computing: Challenges, design aspects, and models for serverless paradigm at the edge. In *The Essence of Software Engineering*; Springer: Cham, Switzerland, 2018; pp. 121–136.
39. Pu, Q.; Ananthanarayanan, G.; Bodik, P.; Kandula, S.; Akella, A.; Bahl, P.; Stoica, I. Low Latency Geo-distributed Data Analytics. *SIGCOMM Comput. Commun. Rev.* **2015**, *45*, 421–434. [CrossRef]
40. Cardellini, V.; Grassi, V.; Lo Presti, F.; Nardelli, M. Optimal Operator Placement for Distributed Stream Processing Applications. In Proceedings of the DEBS '16 10th ACM International Conference on Distributed and Event-based Systems, Irvine, CA, USA, 20–24 June 2016; ACM: New York, NY, USA, 2016; pp. 69–80. [CrossRef]
41. Nardelli, M.; Nastic, S.; Dustdar, S.; Villari, M.; Ranjan, R. Osmotic Flow: Osmotic Computing + IoT Workflow. *IEEE Cloud Comput.* **2017**, *4*, 68–75. [CrossRef]
42. d. Lara, E.; Gomes, C.S.; Langridge, S.; Mortazavi, S.H.; Roodi, M. Poster Abstract: Hierarchical Serverless Computing for the Mobile Edge. In Proceedings of the 2016 IEEE/ACM Symposium on Edge Computing (SEC), Washington, DC, USA, 27–28 October 2016; pp. 109–110. [CrossRef]
43. Gascon-Samson, J.; Rafiuzzaman, M.; Pattabiraman, K. ThingsJS: Towards a Flexible and Self-adaptable Middleware for Dynamic and Heterogeneous IoT Environments. In Proceedings of the M4IoT '17 4th Workshop on Middleware and Applications for the Internet of Things, Las Vegas, NV, USA, 11 December 2017; ACM: New York, NY, USA, 2017; pp. 11–16. [CrossRef]
44. Thai, M.T.; Lin, Y.D.; Lai, Y.C.; Chien, H.T. Workload and capacity optimization for cloud-edge computing systems with vertical and horizontal offloading. *IEEE Trans. Netw. Serv. Manag.* **2019**, *17*, 227–238. [CrossRef]
45. Deb, P.K.; Misra, S.; Mukherjee, A. Latency-aware horizontal computation offloading for parallel processing in fog-enabled IoT. *IEEE Syst. J.* **2021**, *16*, 2537–2544. [CrossRef]
46. Flores, H.; Su, X.; Kostakos, V.; Ding, A.Y.; Nurmi, P.; Tarkoma, S.; Hui, P.; Li, Y. Large-scale offloading in the Internet of Things. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 13–17 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 479–484.
47. Mostafa, M.A.A.A.; Khater, A.M. Horizontal offloading mechanism for IoT application in fog computing using microservices case study: Traffic management system. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 640–647.
48. Higuchi, T.; Ucar, S.; Altintas, O. Offloading tasks to vehicular virtual edge servers. In Proceedings of the 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW), Monterey, CA, USA, 4–7 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 162–163.
49. Akutsu, K.; Phung-Duc, T.; Lai, Y.C.; Lin, Y.D. Analyzing vertical and horizontal offloading in federated cloud and edge computing systems. *Telecommun. Syst.* **2022**, *79*, 447–459. [CrossRef]
50. Weiser, M. The computer for the 21st century. *Sci. Am.* **1991**, *265*, 94–104. [CrossRef]