

Cloud, Edge, or Both? Towards Decision Support for Designing IoT Applications

Majid Ashouri, Paul Davidsson and Romina Spalazzese

*Department of Computer Science and Media Technology
Internet of Things and People Research Center, Malmö University
Malmö, Sweden
{majid.ashouri, paul.davidsson, romina.spalazzese}@mau.se*

Abstract—The rapidly evolving Internet of Things (IoT) includes applications which might generate a huge amount of data, this requires appropriate platforms and support methods. Cloud computing offers attractive computational and storage solutions to cope with these issues. However, sending to centralized servers all the data generated at the edge of the network causes latency, energy consumption, and high bandwidth demand. Performing some computations at the edge of the network, known as Edge computing, and using a hybrid Edge-Cloud architecture can help addressing these challenges. While such architecture may provide new opportunities to distribute IoT applications, making optimal decisions regarding where to deploy the different application components is not an easy and straightforward task for designers. Supporting designers' decisions by considering key quality attributes impacting them in an Edge-Cloud architecture has not been investigated yet. In this paper, we: explore the importance of decision support for the designers, discuss how different attributes impact the decisions, and describe the required steps toward a decision support framework for IoT application designers.

Keywords—IoT, Edge computing, Cloud computing, Decision support, Application distribution, Quality attributes

I. INTRODUCTION

During the last decade, there has been a trend to utilize Cloud services as hosting infrastructure in a cost effective and dynamic manner. In other words, some of the important functions, such as computation and storage tend to be provided by large centralized data centres. However, the emergence of the Internet of Things (IoT) with devices residing at the edge of the network poses new requirements and issues that challenge Cloud solutions to provide efficient service provisioning [1]. Many IoT applications, such as vehicular safety applications require to be served in a real-time manner with just a few milliseconds latency [2]. Moreover, a rapidly increasing number of connected devices will generate massive amounts of data. For example, it is estimated that the US smart grid will generate 1000 petabytes of data each year, while in 2010 AT&T's network consumed 200 petabytes [3]. In addition, there are several other issues like dynamic scalability, security, privacy, and cost which motivate emerging new technologies alongside of the cloud computing to address these requirements [1][4].

In this regard, the idea of utilizing distributed resources at the edge of the network and providing the required capabilities closer to the source of data has been proposed recently [4][5]. This concept is both known as Edge or Fog computing. Although these terms are often used interchangeably [4], we believe that Edge computing has a broader meaning and includes Fog computing. Indeed, Edge computing refers to all the various kinds of computation and storage capabilities of edge devices that may be used to serve IoT applications, while Fog computing describes an extension of Cloud computing where nodes have a close collaboration with Cloud servers for service provisioning capabilities at the edge of the network [6][7].

Edge computing provides various computation capabilities from the distributed devices at the edge of the network. Utilizing Edge computing impacts the way of processing data compared to the traditional Cloud architecture. To benefit from both Edge and Cloud computing, hybrid Edge-Cloud models have been proposed [8][9]. However, using such a hybrid model adds more complexity when designing IoT applications. IoT applications typically consist of a set of software components that are distributed across the network and have data processing and storage requirements. Adding more options to distribute the components poses challenges regarding how to choose the best solution to deploy application components among all possible combinations. Based on the Edge-Cloud model, an IoT application components can be executed on IoT devices, Edge nodes, and the Cloud. Making a decision on how to properly distribute components is challenging and requires deep analysis.

The application distribution problem in an Edge-Cloud model has been investigated in several studies [10][11][12][13]. They mostly focus on how to efficiently allocate the available resources in the Edge or Cloud to satisfy the received requests from IoT applications in order to maximize the performance of the system with the minimum cost. In other words, they investigate the problem from an infrastructure provider's point of view, in order to utilize their available resources in the optimum way. Instead, here we aim to study the application component distribution problem from another perspective, i.e., from the designer's point of view.

The IoT domain and its applications are extensive and application developers are constantly working on developing

new services or adding features. Therefore, evaluating the different options in Edge-Cloud model at the design time and supporting application and system designers with suggestions would help them to provide services more efficiently taking into account the specific requirements of the individual IoT application. Moreover, the output of the design-time evaluation would help the infrastructure providers to offer more relevant and satisfactory services to the end-users and the designers. To the best of our knowledge, the problem of design-time decision support in a hybrid Edge-Cloud model has not been investigated yet.

To evaluate the different design options in hybrid Edge-Cloud models, the key attributes that influence the decision making should be identified. Previous work mostly focus on few attributes, like energy consumption and latency [10][11][14]. Based on our observation in some real-world use-cases, we found that there are more attributes that impact the decision. Thus, in addition to introducing the concept of design-time decision support, we also identify and describe the relevant key attributes influencing the decision. Finally, we will also propose a series of potential research areas to be investigated in future to pave the way toward design-time decision support. Specifically, the contributions of this paper can be summarized as follows. We:

- Introduce the concept of design-time decision support for IoT application designers in an Edge-Cloud architecture;
- Identify and describe key attributes for IoT application components' distribution in an Edge-Cloud architecture;
- Propose a future research roadmap toward a design-time decision support framework in an Edge-Cloud architecture.

The rest of the paper is organized as follows. The related work is discussed in Section II. In Section III, the detailed architecture of the Edge-Cloud model with explanation of its layers, and application distribution scenarios is presented. We define some important metrics for decision support in Section IV, and then we show its practical usage in some real use cases in Section V. Finally, in Section VI the future research direction will be proposed.

II. RELATED WORK

Edge and Fog computing are relatively new paradigms and related research is still in the early stages. Although some useful studies have been conducted to address the IoT application distribution problem in the Edge-Cloud model, more efforts are required in order to make the proposed contributions mature enough for practical usage. Note that the terms “application mapping” and “application distribution” are used interchangeably in this paper.

Sarkar et al. [14] focuses on a comparison between Fog and Cloud computing by considering simple generalized modelling of delay, power consumption and cost in order to evaluate performance of IoT applications with and without existence of Fog computing. Based on the simple simulation setup, they

argue for the importance of Fog computing to reduce energy consumption, latency and cost, but do not discuss the trade-offs for decision making in the scenarios with conflicting attributes.

By concentrating on the optimization problems, the research presented in [10] [11] [12][13] [15] try to address the application distribution problem from an infrastructure provider point of view. Deng et al. [11] propose an optimization model for power-latency trade-off in a hybrid Fog-Cloud model in order to optimize the workload allocation problem. They use a simple mathematical model of power consumption and latency. Similarly, to minimize the system cost that is a weighted sum of delay and energy consumption with guaranteeing the maximum tolerable delay, Du et al. [10] propose a centralized decision making model to decide where the applications should be processed. Yousefpour et al. [15] and Taneja & Davy [13] propose application distribution solutions for Fog-Cloud scenario based on minimizing IoT nodes delay. Liu et al. [12] propose a multi-objective optimization model for computation offloading. The approach processes requests for application distribution based on the minimizing energy consumption, execution delay, and cost.

In addition, there are other research work focusing on the different aspects of application mapping, e.g. [16][17][18], but they investigate the problem from the point of view of Cloud and Edge computing providers while employing limited attributes.

To sum up, the previous work on the application mapping for IoT systems has focused on the infrastructure provider perspective. Moreover, only a few of the relevant attributes have been studied, typically energy consumption and latency. To consider multiple attributes for design-time application distribution has not been investigated, to the best of our knowledge.

We believe our work can be of help both to IoT software application designers, for which the physical infrastructure available is given, and to IoT system designers, who also decide what computational infrastructure to use, e.g., what processor and how much memory should be in the different nodes of the system. In addition, studying the problem from a designer point of view would also help the above mentioned work to provide more intelligent and satisfactory resource allocation solutions. Moreover, while some existing work consider a limited number of attributes in their decision making problem formulation, we will highlight in the next sections that there are more influential attributes that should be considered to make optimal decisions.

III. EDGE-CLOUD APPLICATION DISTRIBUTION

In this section we first present an Edge-Cloud reference architecture and the details of its layers, and then we discuss different application distribution scenarios through which IoT applications can be deployed.

A. Hybrid Edge-Cloud Reference Architecture

Figure 1 shows the Hybrid Edge-Cloud Reference Architecture that includes three layers: the Cloud Layer, the Edge Layer, and the Thing Layer. As previously mentioned,

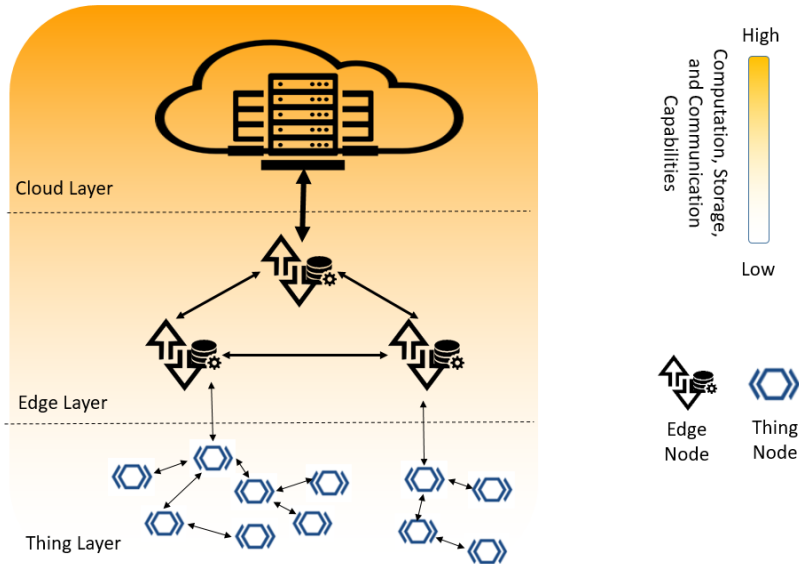


Fig. 1. Hybrid Edge-Cloud Reference Architecture

the Edge Layer includes the so-called Fog and is an intermediate layer between Thing Nodes belonging to the Thing Layer (IoT devices) and the Cloud. The architecture shows a logical view that is simple yet suitable to illustrate the needed concepts for an application distribution model.

In the following, we describe the different roles and responsibilities of each layer of the architecture in more detail. The separation among the layers is based on basic capabilities, i.e., sensing and actuation, storage, processing, and communication.

Thing Layer: The Thing Nodes included in this layer are devices that typically have very limited computational and storage resources that enable them to host very simple applications such as sensing phenomena, actuation based on certain events, and sending and receiving data.

Edge Layer: This layer consists of nodes with more computational, storage, and communication capabilities than the Thing Layer (but still limited) that enables them to manipulate, analyse, and dispatch data, interact with the Cloud and Thing Layers, and host IoT applications or components located at the network edge. The role of Edge Nodes can be played by various devices, e.g., gateways, access points, routers, switches, local servers, smartphones and modern vehicles.

It is worth to notice that some devices can play different roles. For example, a traffic light can act at the same time both as a Thing Node that senses the environment and actuates, and as an Edge Node that aggregates data, processes it, and hosts application components.

Cloud Layer: The Cloud Layer consists of several high-end servers and data centres with massive computational, storage and communication capabilities that enables them to host various kinds of IoT applications with high physical resources demand.

Adding the Edge Layer for hosting IoT applications to the Cloud Layer would be beneficial for several reasons: it enables Cloud providers to provide their services with a higher quality, e.g., like reducing latency; it enables application designers to develop more efficient and cost effective applications addressing end user requirements; it reduces the amount of data communication through the network thus supporting network infrastructure providers; finally, it supports privacy and provides a better user experience to the end-users. Here we concentrate on the designer point of view, although there might be similarities with other views.

To be able to benefit of an Edge-Cloud solution, an appropriate distribution of the IoT application components to the different layers is critical. A wrong distribution might decrease the performances and increase the cost of design and hosting of the applications. In the following, we will describe different application distribution options and in Section V we will illustrate the problem of distributing application components through some real world use cases.

B. Application Distribution Model: Three Guidelines

IoT applications are formed by several components that communicate in order to serve a specific purpose. Emerging enablers in the IoT domain such as Cloud and Edge computing technologies provide new options to design and develop more efficient applications. Thus, IoT designers should be guided about how to distribute applications and receive support to make tradeoffs analysis to identify the best solution that can range from *deploying everything at the Edge*, or *everything at the Cloud*, or adopting a *hybrid solution*. Here, we briefly discuss how different components of an application can be mapped to the different layers of our described architecture.

1) *Distribution of application components on the Thing Nodes.* Since sensing and actuation functions are performed by the Thing Layer in the architecture, it is simple and straightforward to distribute components with these kind of capabilities and/or needs on Thing Nodes.

2) *Distribution of application components on the Edge Nodes.* This option is suitable in two cases: (i) when one or more components of an application can be run locally without any interaction with the Cloud; (ii) when Edge Nodes act as data proxies and computing offloading nodes which collaborate with the Cloud to serve a service. Since processing, storage and communication capabilities have some limitations in the Edge Nodes, distributing components requiring huge resources to this layer may not be the optimal choice.

3) *Distribution of application components on the Cloud data centres.* Deploying applications' components on the Cloud data centres is a suitable option when they have massive and elastic processing and storage requirements or in case of need to collect data from various distributed locations.

Based on the three above mentioned mapping possibilities, it is clear that an IoT application can be distributed in a number of different ways on the Hybrid Edge-Cloud Architecture. It is worth mentioning that in real world scenarios, mapping a component to a specific layer is not a deterministic decision. Designers might express a preference or priority about where to map a component, and this should be taken into consideration while computing possible solutions. In order to find possible solutions options, several quality attributes should be considered and measured. In the following section important attributes in the Edge-Cloud Architecture will be discussed.

IV. DECISION SUPPORT ATTRIBUTES

In this section we describe several key quality attributes that are important to consider for supporting IoT designers in their component distribution decisions. The details of each attribute and its related metrics will be the subject of future studies. The extracted attributes are from a designer's point of view, and what is that influence designer decisions regarding the application distribution. Note that most of the proposed attributes are also relevant from an infrastructure provider point of view. The selection of attributes is based on both a review of the literature and a number of real world IoT applications.

Response time / actuating latency: Response time is the duration time from when an IoT device makes a request, possibly as a consequence of user interaction with the IoT system, until the IoT device receives a response. The term actuating latency is the duration time from when an event is detected until it is properly actuated to the target IoT devices.

Supporting real-time responses or real-time actuation has been considered one of the main goals behind proposing Edge computing to enhance IoT applications [4][7]. In this regard, measuring real-timeliness of applications and allocating resources based on the related metrics has been an interesting subject studied by several state of the art works [10-15].

Energy consumption: Energy consumption as a decision making attribute is popular among research works in IoT and Edge computing, especially when the workload or resource allocation problem is considered as a main target. The energy consumption attribute has been studied from different point of views. For example from the IoT devices point of view, the main target is designing services consuming less energy in the end devices and prolonging battery life [19]; from resource allocation point of view the main concern is how to reduce the amount of energy required by processing and storage resources in order to reduce the cost [10][11]; and from governmental or eco-friendliness view the total energy consumptions of the system would be the desired attribute. This diversity shows that depending on the point of view, different kinds of metrics can be used to evaluate the application component. To measure the amount of consumed energy, it is common to divide it into processing, storage and communication energy consumption parts [14].

Resource usage: Resource usage can be evaluated in terms of bandwidth, computation and storage. Considering and measuring this attribute is critical in evaluating IoT applications, since it has also a direct and important impact on the other aspects like the cost, performance and energy consumption of the system. Bandwidth, computation and storage attributes have been used in several works like [13][18] related to the hybrid Edge-Cloud model. However, they mostly consider these attributes as available resources that should be allocated to the different applications in an optimum way. Comparing the amount of resource usage in different solution configurations has not been considered yet, to the best of our knowledge.

Accuracy: The accuracy of the service functionality measures the degree of proximity to the user's actual values when using a service compared to the expected values [20]. In Edge-Cloud model this is a critical aspect to measure since it makes a trade-off between processing data distributed in edge devices or processing them centrally with more resources in the Cloud. By processing applications in the edge, it would make it possible to reach lower transmission latency and reduction in data transmission, but processing and storage resources are limited and it narrows accessing data in terms of time and location that would impact the accuracy of the results. Moreover, the level of required accuracy varies in different applications; for example, in a smart home improving energy consumption and its performance in 90% accuracy might be acceptable while it is not valid for most safety applications like accident prevention. Unfortunately, this attribute has not been considered as a key attribute in the Edge-Cloud application distribution works yet.

Availability: Availability is the percentage of time a customer can access a service [20]. Availability is a very important attribute for many use cases especially for safety, control systems, and industrial IoT applications. Considering this attribute while evaluating the Cloud Computing services is very common to evaluate the Cloud Computing services, but it has not been considered among the state of the art application distribution solutions in the Edge-Cloud model.

Security and privacy: Information security and privacy are common concerns for every end user and designer. The emergence of Cloud computing ignites a discussion of how Cloud providers ensure secure and private hosting services while they have an access to the data and host other services as well. This issue would be more challenging by extending a centralized Cloud to decentralized Edge Nodes, as investigated in [21][22]. Security and Privacy are multi-dimensional attributes that can be measured and expressed by different parameters based on the specific domain and designers concerns.

Costs: Generally, cost is measured in terms of CAPital and OPerational EXPenses (CAPEX and OPEX). In the form of designer point of view, it can be divided to the system design and implementation costs, and operational costs such as the charging cost (for the allocated resources in Edge, Cloud infrastructure) and product maintenance costs. In order to reduce the capital and operational costs, resource hosting solutions like Cloud and Edge computing try to offer attractive suggestions for service providers.

Due to maturity and practical deployment of Cloud computing, currently various cost models and metrics have been proposed to utilize it [23]. However, these models should be extended and improved to consider integration with Edge computing, by mean of which a designer decides about the affordable solution to rely on.

Elasticity: “Elasticity is the degree to which a system is able to adapt to workload changes by provisioning and deprovisioning resources in an autonomous manner, such that at each point in time the available resources match the current demand as closely as possible” [25]. This is mostly defined by two attributes: mean time taken to expand or contract the service capacity, and maximum capacity of service. The capacity is the maximum number of compute units that can be provided at peak times [20]. In the dynamic environments with rapid variations in required resources, elasticity would be an important parameter that should be considered by designers, since elasticity degree has direct impact on cost and performance attributes like accuracy, availability and latency of the system.

Scalability: Scalability is the ability of the system to sustain increasing workloads by making use of additional resources. It is a prerequisite for elasticity, but it does not consider temporal aspects of how fast, how often, and at what granularity scaling actions can be performed.

There are several additional candidate attributes, but they can typically be reduced to a combination of some of the more basic ones that we have selected and described above. For instance, when deciding how to distribute the components of an IoT system, the attribute “user experience” can to a significant degree be captured by a combination of basic attributes like response time, availability, and privacy. However, our selection is still preliminary and will be further validated.

V. USE CASES SCENARIOS

In this section we describe and analyse two real world use cases. The choice is based on the variety of system design

requirements to illustrate the importance of decision support in a hybrid Edge-Cloud model. Based on the key attributes of the two use case scenarios, we show how the system components can be mapped to different layers. Obviously, the importance of attributes is dependent on deployment conditions and application requirements and would be different in other scenarios. However, here the main goal is to show the challenges of the decision making when there are multiple key attributes.

A. Smart Traffic Light

Smart Traffic Light (STL) typically is a part of a more complex Intelligent Transportation System and Connected Vehicles. Abstractly, an STL system contains several distributed components: traffic sensing, data aggregation, real-time cycle scheduling, cycle display, and overall system long-term optimization.

The use case concerns scheduling and cycle management of traffic lights. It includes different software components that can be distributed in different ways. Therefore, utilizing a hybrid Edge-Cloud solution and considering traffic light nodes as Edge nodes can be accounted as a design option. For example, the following distribution would be possible: Traffic sensing and cycle display components could be executed on Thing nodes such as magnetic sensors and traffic light display; data aggregation and real-time cycle management components could be run on Edge nodes like traffic light controllers or local servers; finally, components such as data aggregation, real-time cycle scheduling, and long-term optimization components can be allocated to Cloud data centres. For this use case, the data aggregation and real-time cycle management components are more challenging for the designer and require decision support to find the best option.

To show the role of attributes in the decisions let us consider the traffic light controller device as an Edge node and candidate to host the real-time cycle management and data aggregation components. To make accurate decisions and to provide a green wave, it is necessary to collect information from other intersections and controllers in addition to the data from the local sensors. Thus, in this case the accuracy and availability of the system should be evaluated since different level of data is accessible in the Edge compared to a centralized solution. If energy consumption is also considered as an important factor, the amount energy by propagating the collected data of a traffic light node to several other nodes should be measured, since it may even increase the total energy consumption. Latency is another important attribute, but there is not a hard real-time constraint for this service. The available computation resources and data synchronization between adjacent controllers influence the latency. The number of STL nodes and their density also impacts decision in terms of cost and latency. For example, if there are several isolated STL with a few crossing cars, it might be an option to run all components in a STL node without communication with other entities. Moreover, employing dependable security mechanisms is critical, and if sensing is performed by image processing methods, privacy is also an attribute that should be considered.

B. Smart Video Surveillance

Nowadays, Smart Video Surveillance (SVS) systems are used in a variety of applications like object tracking, object recognition, ID re-identification, customized event alerting, and behaviour analysis [26]. Although there has been a great improvement of the enabling technologies so that some of the video surveillance applications can be performed locally, they can also be performed on other entities with more processing resources (and to a lower cost) in order to provide better quality of service by using video processing, computer vision algorithms and pattern recognition [27][28]. Moreover, since the generated data consists of local and geo-constrained data streams, it is a possibility to consider a Cloud-Edge solution.

Let us consider the use case of red light violation detection and accident prevention in the intersections. To detect abnormal driving and red light violation, one way is to track objects via the SVS. This use case consists of several components: movement capturing, object tracking, data compression, abnormal behaviour detection, warning management, warning display, and overall system long-term optimization. An example of application distribution for this use case is the following. The sensing and actuating functions, i.e. movement capturing and warning display, can be deployed in the Thing layer on a camera and a special display device. Long-term algorithm optimization is also expected to be placed on the Cloud, but the other components like object tracking, data compression, abnormal behaviour detection, and warning management can be executed on the Edge nodes like a camera or local servers, or executed on Cloud data centres.

To find the best option for distributing each component, a number of attributes should be considered. In this use case, there is an obvious real-time requirement with a high degree of accuracy. Thus, on the one hand considering an Edge solution with low communication latency by eliminating best-effort impacts of IP networks would be a reasonable option. However, on the other hand, it may also increase computation latency in comparison to Cloud computing, and reduce the event detection accuracy. Moreover, streaming a massive amount of data to the Cloud in the form of high resolution images, imposes considerable energy consumption and bandwidth cost for some extra processing features. Providing a high degree of privacy and security is also an influential factor in this scenario. Although the overall cost is always an important attribute, compared to the importance of saving lives it may have less impact on the final decision. Having acceptable performance in a rapid change of traffic, and its scalability are the other key attributes that impact the decision.

VI. FUTURE RESEARCH DIRECTIONS

To realize the decision support for IoT designers in the Hybrid Edge-Cloud model, several research directions need further investigation. In this section, we will summarize these opportunities for further research including attribute operationalization, decision support framework, and resource allocation improvements.

A. Operationalization of Attributes

As discussed in previous sections, an accurate decision support would be realized by considering a number of important attributes. To be effectively used in decision support systems, the attributes should be defined so that is possible to measure and evaluate them through metrics. The attributes discussed in Section IV are different in nature and have been used differently in the literature. Some are easy to model and measure, and have been referred widely while some are more subjective and difficult to measure mathematically. In this regard, among the discussed attributed latency and energy consumption have attracted more attentions in the recent works [10][11][14]. The importance of these attributes for resource allocation, their simplicity for modelling mathematically, and the ability to provide generalized model for different application are some of the reasons that persuaded researchers to work on them. In addition, resource usage, and cost are the other attributes commonly used in recent works [13][14]. Unfortunately, for the remaining attributes there has been less attention in spite of their importance for application distribution purposes. More effort is needed to propose measurable metrics for those attributes in order to be used by the decision support frameworks. It is also worth mentioning that, it would be relatively straightforward to mathematically model some attributes like energy consumption and latency, while it would be more complex to propose a model for some others that are more subjective like security and privacy.

B. Proposing a Decision Support Framework for Application Distribution

Having measurable metrics in hand would not be sufficient to support designer decisions. To offer a comprehensive solution, a general decision support framework containing the proposed attributes is required. The framework should be able to take the designer requirements and weight the attributes based on their general importance and the end user preferences for the specific application or component. Then a ranking algorithm should process the attributes values and weight to suggest a ranking of solution options. The decision support framework output would be a list of solutions on which application components can be lunched along with some guidelines for designers. The framework should also be flexible enough to work with new attributes or new metrics for specific attributes.

The kind of problems that different attributes (criteria) impact the decision can be defined as multiple criteria decision making (MCDM) problems. There are different approaches toward solving these problems among which Multiple Attribute Utility Theory (MAUT), outranking, and Analytic Hierarchy Process (AHP) are the fundamental methods. There have been many works in the literature using these methods for solving multiple criteria problems, such as [20] and [29]. In [20] a framework for selecting Cloud computing solutions based on AHP approach has been proposed. In [29] an AHP based approach has been used to evaluate the multi agent systems for different applications.

C. Resource Allocation Improvement

Resource allocation mechanisms in the Hybrid Edge-Cloud architecture have been investigated by several research works as described in Section II. Resources allocation is usually performed by considering some attributes that are important from a Cloud/Edge provider point of view like energy consumption and latency along with some inputs from users like required resources or latency threshold. Adding the designer point of view and considering designer needs and preferences about the solutions would result to more satisfactory service provisioning for the users. For example, let us assume that a designer can provide preferences of options for application components as weighting percentage, and it is estimated that a component has a priority of 70% to be run on the Edge. Thus, by taking designer's preferences into account, resource allocation algorithms will provide more satisfactory services by addressing the designers and end users requirements.

VII. CONCLUSION

In this paper we introduced the idea of design-time decision support for developers of IoT applications using a hybrid Edge-Cloud architecture. Using an abstract hybrid Edge-Cloud Reference Architecture with three layers, we described how different application components can be distributed across the different layers of the architecture and discussed the challenges involved when distributing IoT application components. We identified the key attributes that should be considered when deciding how to distribute the components. We then analysed how these attributes impact design decisions through two use cases. Finally, we discussed some research directions towards a decision support framework.

We believe that our work can be a basis and provide a roadmap for future research in application distribution for hybrid Edge-Cloud architectures. Our plan for future investigations includes: to devise a decision support framework by considering the identified key quality attributes; to propose an IoT application component distribution method to effectively support designers.

ACKNOWLEDGMENTS

This work is partially financed by the Knowledge Foundation through the Internet of Things and People research profile (Malmö University, Sweden).

REFERENCES

- [1] M. Chiang, T. Zhang, "Fog and IoT: An Overview of Research Opportunities," *IEEE Internet Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.
- [2] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 4, pp. 584–616, 2011.
- [3] N. Cochrane, "US smart grid to generate 1000 petabytes of data a year." <<http://www.itnews.com.au/news/us-smart-grid-to-generate-1000-petabytes-of-data-a-year-170290#ixzz458ValTi6>> [Published: March 23, 2010].
- [4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [5] W. Yu *et al.*, "A Survey on the Edge Computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2017.
- [6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of things," *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ser. MCC'12. ACM*, pp. 13–16, 2012.
- [7] F. Bonomi, R. Milito, P. Natarajan, J. Zhu, N. Bessis, C. Dobre, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments, New York, NY, USA:Springer*, pp. 169–186, 2014.
- [8] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, M. Nemirovsky, "Key ingredients in an IoT recipe: Fog Computing, cloud computing and more fog computing" in *IEEE 19th Int. Workshop (CAMAD) Computer Aided Modeling and Design of Communication Links and Networks*, pp. 325–329, 2014.
- [9] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G. J. Ren, "Foggy clouds and cloudy fogs: A real need for coordinated management of fog-to-cloud computing systems," *IEEE Wireless Communication Magazine.*, vol. 23, no. 5, pp. 120–128, 2016.
- [10] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation Offloading and Resource Allocation in Mixed Fog/Cloud Computing Systems with Min-Max Fairness Guarantee," *IEEE Transactions on Communications*, vol. 66, pp. 1594–1608, 2017.
- [11] R. Deng, R. Lu, C. Lai, and T. H. Luan, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171–1181, 2016.
- [12] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multi-objective Optimization for Computation Offloading in Fog Computing," *IEEE Internet Things Journal.*, vol. 5, no. 1, pp. 283–294, 2017.
- [13] M. Taneja and A. Davy, "Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm," *IFIP/IEEE International Symposium on Integrated Network Management.*, pp. 1222–1228, 2017.
- [14] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the Suitability of Fog Computing in the Context of Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, 2015.
- [15] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog Computing: Towards Minimizing Delay in the Internet of Things," *Proc. - 2017 IEEE 1st Int. Conf. Edge Comput.*, pp. 17–24, 2017.
- [16] N. Verba, K.-M. Chao, A. James, J. Lewandowski, X. Fei, and C.-F. Tsai, "Graph Analysis of Fog Computing Systems for Industry 4.0," *IEEE 14th Int. Conf. E-bus. Eng.*, pp. 46–53, 2017.
- [17] J. P. D. Comput, R. Mahmud, S. Narayana, and K. Ramamohanarao, "Quality of Experience (QoE)-aware placement of applications in Fog computing environments," *Journal of Parallel and Distributed Computing*, 2018.
- [18] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Resource provisioning for IoT application services in smart cities," *13th Int. Conf.*

Netw. Serv. Manag., pp. 1–9, 2017.

- [19] M. Ashouri, H. Yousefi, J. Basiri, A. M. A. Hemmatyar, and A. Movaghar, “PDC: Prediction-based data-aware clustering in wireless sensor networks,” *J. Parallel Distrib. Comput.*, vol. 81–82, pp. 24–35, 2015.
- [20] S. K. Garg, S. Versteeg, and R. Buyya, “A framework for ranking of cloud computing services,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 4, pp. 1012–1023, 2013.
- [21] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, “Fog Computing for the Internet of Things: Security and Privacy Issues,” *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, 2017.
- [22] I. Stojmenovic and S. Wen, “The Fog Computing Paradigm: Scenarios and Security Issues,” in *Proc. 2014 Federated Conf. Comp. Sci. and Info. Sys.*, vol. 2, pp. 1–8, 2014.
- [23] M. Al-Roomi, S. Al-Ebrahim, S. Buqrais, and I. Ahmad, “Cloud Computing Pricing Models: A Survey,” *Int. J. Grid Distrib. Comput.*, vol. 6, no. 5, pp. 93–106, 2013.
- [24] S. K. Garg and R. Buyya, “Green Cloud computing and Environmental Sustainability,” in *Harnessing Green IT: Principles and Practices*, S. M. a. G. G. (eds), Ed. UK: Wiley Press, pp. 315-340, 2012..
- [25] N. R. Herbst, S. Kounev, and R. Reussner, “Elasticity in Cloud Computing: What It Is, and What It Is Not,” in *10th Int. Conf. Auton. Comput.*, pp. 23–27, 2013.
- [26] V. Tsakanikas and T. Dagiuklas, “Video surveillance systems-current status and future trends,” *Comput. Electr. Eng.*, pp. 1–18, 2017.
- [27] A. Botta, W. De Donato, V. Persico, and A. Pescape, “On the integration of cloud computing and internet of things,” *Proc. Int. Conf. Futur. Internet Things Cloud*, pp. 23–30, 2014.
- [28] N. Chen, Y. Chen, Y. You, H. Ling, P. Liang, and R. Zimmermann, “Dynamic urban surveillance video stream processing using fog computing,” *Proc. IEEE 2nd Int. Conf. Multimed. Big Data*, pp. 105–112, 2016.
- [29] P. Davidsson, S. Johansson, and M. Svahnberg, “Using the analytic hierarchy process for evaluating multi-agent system architecture candidates,” *Agent-Oriented Softw. Eng. VI Lecture Notes in Computer Science*, vol.3950, pp. 205–217, 2006.