2018 ACM/IEEE 13th International Symposium on Software Engineering for Adaptive and Self-Managing Systems

Decentralized Self-Adaptive Computing at the Edge

Mirko D'Angelo
Linnaeus University
Växjö, Sweden
mirko.dangelo@lnu.se

ABSTRACT

Nowadays, computing infrastructures are usually deployed in fully controlled environments and managed in a centralized fashion. Leveraging on centralized infrastructures prevent the system to deal with scalability and performance issues, which are inherent to modern large-scale data-intensive applications. On the other hand, we envision fully decentralized computing infrastructures deployed at the edge of the network providing the required support for operating data-intensive systems. However, engineering such systems raises many challenges, as decentralization introduces uncertainty, which in turn may harm the dependability of the system. To this end, self-adaptation is a key approach to manage uncertainties at runtime and satisfy the requirements of decentralized data-intensive systems. This paper shows the research directions and current contributions towards this vision by (*i*) evaluating the impact of the distribution of computational entities, (*ii*) engineering decentralized computing through self-adaptation and, (*iii*) evaluating decentralized and self-adaptive applications.

ACM Reference Format:
Mirko D'Angelo. 2018. Decentralized Self-Adaptive Computing at the Edge. In *SEAMS '18: SEAMS '18: 13th International Symposium on Software Engineering for Adaptive and Self-Managing Systems , May 28–29, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3194133.3194160

1 INTRODUCTION

With the growing demand for real-time data originating from Internet-connected devices, the number of requests hitting today's computing infrastructures goes beyond what is manageable for operations and affordable for management. According to Cisco's annual mobile data forecast [6], traffic has grown 18-fold over the past 5 years. Moreover, by 2021, there will be 11.6 billion mobile-connected devices, exceeding the world's projected population at that time. Indeed, future applications will be characterized by data-intensive tasks which need to process, move and manipulate extremely large volumes of data and devote most of their processing time to input/output operations. As an example of this phenomenon, consider that every 60 seconds more than 11 million of instant messages are exchanged over the internet and more than 98000 tweets are produced.

SEAMS '18, May 28–29, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5715-9/18/05.
https://doi.org/10.1145/3194133.3194160

Existing ongoing research on big data is twofold [23]: while the infrastructure-centric view focuses on architectures, computing, networking and dependability engineering, the data-centric view focuses on properties and values generated by the data. In this paper, we mainly focus on the infrastructure-centric view of data-intensive systems to guarantee for scalability and performance.

Computing infrastructures aim at providing the hardware and services that applications use to build their functionalities on. In the currently adopted computing infrastructures (e.g., Cloud Computing), computations are executed remotely on servers deployed at the center of the infrastructure. Even though these solutions work well today, they fail when considering myriads of devices interacting each other by exchanging micro-data. In fact, many data-intensive applications, such as IoT data-streaming and intelligent transportation systems, cannot tolerate latency issues associated with the Cloud Computing paradigm, where data is sent back and forth between client and server [2]. Other academic or industry solutions (e.g., Fog Computing) distribute the computing infrastructure close to the users with the aim of addressing this problem. However, in these paradigms devices are still considered to be simple portals to reach the infrastructure and applications are usually managed by a logically centralized point of coordination, which may hinder scalability and performance.

Over the last decade, devices have become increasingly powerful, with significant processor power and memory. However, these computing capabilities are nowadays often underutilized. In fact, in order to fully benefit from the mobile nature of modern communications and the increasingly computational power available at the edge of the network, we require a radical shift towards computing paradigms that fully reflect the network-based perspective of the execution environment [8]. To this end, we adhere to the Edge Computing paradigm [19], and aim at pushing applications, data, and services away from central nodes to the edge of the network.

Engineering such systems is challenging, as decentralization introduces uncertainty, which in turn may harm the dependability of the system. To this end, self-adaptation proved to be an effective approach to manage uncertainties at runtime and satisfy the requirements of a decentralized data-intensive computing environment. In this paper, we discuss our research directions and current contributions towards the actualization of this vision. In particular, (*i*) we demonstrate that the distribution of only the computing infrastructure is not sufficient to support dependable data-intensive applications and that decentralized management policies are needed, (*ii*) we make a step towards engineering a fully decentralized computing infrastructure and show how to manage uncertainties through self-adaptation, and (*iii*) we propose a simulation-based approach to evaluate decentralized self-adaptive applications.

The paper is organized as follows. In Section 2 we show our research vision and outline important challenges towards its actualization. In Section 3 we present our current contributions. Finally, in Section 4, we draw conclusions and present a planned timeline for research completion.

2 RESEARCH VISION AND CHALLENGES

Computing infrastructures aim at providing the hardware and services that data-intensive systems use to build their services on. It is possible to classify them along different dimensions, namely: architectural style, location of services, physical and logical model, support for client and server mobility and performance [8].

Cloud Computing (CC) heavily rely on distributed processing and available bandwidth from the peripheral devices to the central backed server. In CC most of the data is sent to the central server to be processed, leaving peripheral devices as simple portals to reach the cloud. Mobile Cloud Computing (MCC) exploits Cloud Computing (CC) to bring computation resources to mobile users. Differently from CC, MCC considers various mobile-related factors like offloading strategies, mobility and device energy [21]. However, in both CC and MCC computations are executed on distant servers. Hence, the centralized management of the resources and the inevitable problem of performance make these computing architectures not appropriate in modern scenarios. Mobile Edge Computing (MEC) [24] deals with MCC issues by offering CC capabilities on dedicated servers at the edge of the network, like gateways or cellular base stations. Similar computing paradigms are Fog Computing (FC), which aim at supporting Internet of Things (IoT) applications by using routers to carry out computations [2] and Cloudlets, which can be viewed as a "data center in a box", with no hard state, whose goal is to bring the cloud closer to the user [17].

Even though in some of these paradigms performance (e.g., latency) issues are addressed, devices are still considered to be simple portals to reach the infrastructure. Hence, scalability, to accommodate the growing number of connected devices, can only be addressed by adding more servers or edge servers near the users. Moreover, in these infrastructures, applications are usually managed by a logically centralized point of coordination. This is the case of Cloud and Fog gateways (or controllers), which orchestrate the applications' data-flow among services. Leveraging on centrally managed infrastructures prevent the system to deal with scalability and performance issues, which are inherent to modern large-scale data-intensive applications.

To this end, our research vision is to allow the creation of a P2P computing environment at the edge of the network. Open-world computing paradigms such as Volunteer Computing [7] have already been proposed. However, their application is typically focussed on creating collaborative cloud solutions [18] and usually adopt centralized coordination techniques [1, 13]. On the other hand, our idea is to adopt both a distributed computing infrastructure and decentralized management policies.

Engineering this type of system raises several challenges, including: (*i*) the complexity caused by the potentially high number of entities and the fully decentralized nature of the environment, (*ii*) the unpredictable variability of the environment (e.g., entities join/leave the network and/or change their quality attributes) and,

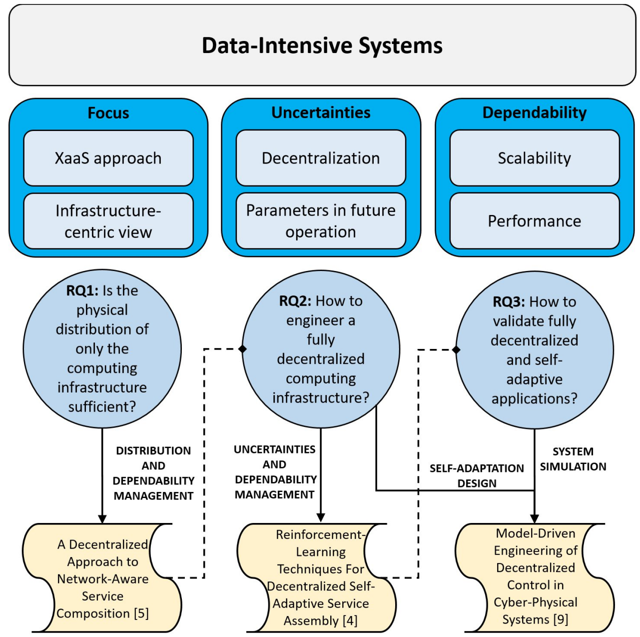

Figure 1: Research Vision and Current Contributions

(*iii*) the lack of global knowledge. Such challenges are categorized as *uncertainties at runtime* [11, 16] and can affect the dependability of the system. To this end, self-adaptation proved to be an effective approach to manage uncertainties at runtime and successfully achieve functional and non-functional requirements of data-intensive applications.

In order to deal with the envisioned system, it is important to have a deep understanding of the underlying uncertainties. To this end, we adhere to the taxonomy presented in [16], which classifies the type of uncertainties along three dimensions: (*i*) *location*, which refers to the place where the uncertainty manifests, (*ii*) *level*, which indicates the uncertainty's spectrum, between deterministic knowledge and total ignorance and, (*iii*) *nature*, which refers to whether the uncertainty is due to the imperfection of the acquired knowledge (i.e., epistemic uncertainty) or is due to the inherent variability of the phenomena (i.e., aleatory uncertainty).

In this work we deal with two types of uncertainties (see Figure 1): uncertainty due to decentralization and uncertainty of parameters in future operation. Uncertainty due to decentralization is a structural uncertainty of epistemic nature, and it is caused by the lack of global knowledge in systems where there is not a central entity that has complete control over the actions of other entities. This is the case of our envisioned system where, differently from the controlled computing infrastructures, the knowledge is scattered among different entities. Uncertainty of parameters in future operation is a parametrical uncertainty of epistemic or aleatory nature. Since self-adaptive systems must consider the behavior of the system in its future operation to be able to achieve certain objectives, uncertainty regarding future system's states arise. The lack of global knowledge or/and the inherent variability of the acquired data could be the source of this uncertainty. For both the uncertainties, we limit our scope to uncertainties of the first order level, which is the case where there is the lack of knowledge about something but entities are aware of such lack (i.e., known uncertainty).

Besides functional requirements, uncertainties also affect the dependability attributes of a system. In this work (see Figure 1) we focus on two dependability attributes: scalability and performance. Scalability is the property that our computing infrastructure must provide in order to handle the growing amount of entities connecting to the environment. Achieving scalability under decentralization is hard since coordination among entities could cause communication overhead within the environment, which in turn might degrade the whole system's performance – e.g., throughput, latency, and response time. Hence, engineering scalable data-intensive applications that guarantee a high level of performance is challenging.

2.1 Problem Formulation

Referring to Figure 1, our research mainly focuses on:

- The investigation of the infrastructure-centric [23] view of dependable data-intensive systems. This topic includes architectures, computing, networking and dependability engineering. The other view that we do not deal with is the data-centric view [23], whose focus is on properties and values generated by data.
- The adoption of a service-oriented approach to data-intensive systems with the aim of linking computational resources. The complexity of integrating such diverse resources can be tackled by adopting the everything-as-a-service (XaaS) abstraction to uniformly represent physical things, hardware resources and software applications, irrespectively of their specific nature [3].

To this extent, the following research questions are investigated (see Figure 1):

RQ1 Is the physical distribution of only the computing infrastructure sufficient to satisfy the scalability and performance requirements of data-intensive applications?

RQ2 How to engineer a fully decentralized computing infrastructure at the edge of the network?

RQ3 How to validate fully decentralized and self-adaptive applications?

Next section, will discuss our current contributions towards our research vision.

3 CURRENT CONTRIBUTIONS

Figure 1 illustrates our current contributions towards the aforementioned research directions. Each work allows for analyzing one or more aspects characterizing the research questions (solid lines) and acts as an enabler for other research questions (dotted lines). Next subsections address each research question individually by discussing achieved results.

3.1 Decentralized Computing

By considering **RQ1**, we started investigating whether the distribution of only the computing infrastructure is sufficient to satisfy the scalability and performance requirements of data-intensive applications [5]. With the recent widespread of data-intensive applications, the composition of large numbers of services into a coherent system is an important research area [3]. In this work, we consider a scenario where the computational entities (i.e., the services) needed to compose specific data-intensive applications, specified as workflows, are distributed in the environment. We designed a decentralized and self-adaptive network-aware service composition approach dealing with the composite service completion time. In particular, we took into account both the response time and the impact of network latency. Self-adaptation plays a central role in the devised composition strategy. In fact, given the intrinsic dynamism of the system, the protocol was designed to adapt according to changes in both the set of available resources and the nodes topology.

We explicitly considered the case of centrally and decentrally orchestrated services, and the different impact these two models have on the network-aware procedure for service composition. In the former scenario a centralized orchestrator manages the data-flow among the composite services while, in the latter case, the data-flow management is completely decentralized and managed in a P2P fashion by the services. By comparing these data-flow management strategies, we aimed at verifying whether the distribution of only the computational entities in the environment is sufficient to satisfy the response time and scalability requirements of data-intensive applications. We evaluated the proposed approaches over a large network by exploiting the event-driven engine of PeerSim [14]. Simulation experiments show that the centralized model simplifies the workflow management, but introduces additional delays caused by the indirect interaction between consecutive services, and may suffer from the typical problems of a centralized solution (e.g., bottleneck node, single point of failure).

We learned that the distribution of only the computational entities is not sufficient to satisfy the response time and scalability requirements of data-intensive applications. In particular, besides the bottlenecking problem, centralized data-flow orchestration solutions bring poor performance and fail to satisfy strict requirements. We noticed approximately 12% of performance degradation with respect to the decentralized orchestration model. Moreover, we noticed that decentralized orchestration was able to scale and react quicker to dynamic changes in the network. This study confirms the importance of adopting decentralized data-flow management techniques to guarantee scalability and performance in data-intensive scenarios: while centralized management solutions on distributed computing infrastructures are appropriate and effective in controlled environments (such as Cloud or Fog computing), they might fail when dealing with large-scale distributed environments with strict performance requirements.

3.2 Self-Adaptation

According to the lesson learned from **RQ1**, **RQ2** aims at investigating how to engineer a fully decentralized computing environment at the edge of the network. We started exploring this problem in a previous work [4], where we tackle this challenge by considering data-intensive applications modeled as composite services. This work further extends the scenario introduced in [5] by considering load-dependent quality attributes, and a general QoS model to manage multiple performance requirements. In fact, besides response time, other performance attributes must be considered when evaluating data-intensive applications (e.g., throughput).

Approaches that mostly assume static data are inadequate for the type of applications and environment we are considering. In fact, it is important to consider the behavior of the system in its future state to be able to deal with its requirements. This is even harder to do in distributed and decentralized settings since there is not a single entity where it is possible to aggregate the data and coordinate the actions. Hence, two types of uncertainties must be tackled: uncertainty due to decentralization and uncertainty due to the need to consider parameters in future operation.

We managed uncertainty due to decentralization by instantiating a self-adaptation engine architected according to the well-known MAPE-K model. In particular, we adopted the information sharing pattern where each entity self-adapts locally by implementing its own MAPE-K loop, but requires information from other entities in the system [22]. To overcome the lack of global knowledge and deal with the uncertainty of parameters in future operation, we adopted a reinforcement learning approach to make each entity able to dynamically learn the most effective service composition rule to be followed. We evaluated the designed system in different dynamic situations by exploiting simulations. Results show that our system is able to build and maintain in a fully decentralized way an assembly of services that, besides functional requirements, is able to guarantee a good overall performance for data-intensive applications in the environment.

We learned that self-adaptation is indispensable to cope with unpredictably variable decentralized computing environments. Moreover, learning strategies are able to predict the system's future state without the need for global knowledge and play a key role in modern data-intensive applications.

3.3 Validating Decentralized Self-Adaptation

The main findings of **RQ1** and **RQ2** are decentralization and self-adaptation, respectively. Finally, **RQ3** aims at providing a contribution on how to evaluate applications that are inherently decentralized and self-adaptive. To deal with the evaluation of such class of systems we apply a systematic empirical inquiry [12]. This method allows for the evaluation of particular solutions by providing a rigorous approach for (*i*) objectives definition, (*ii*) experiments setup, (*iii*) data collection and analysis, and (*iv*) results presentation. Experiments are done through simulation, which allows for assessing design choices before the system is actually put into operation, as well as for evaluating several different performance attributes of interest, including response time, latency and adaptation time.

When it comes to engineer self-adaptive systems and simulate their behavior, there is the lack of tools which allow defining a single and coherent process of both managing and managed subsystem simulation. This is a crucial need in nowadays computing scenarios since applications typically involve cyber-physical entities (e.g., vehicles, smart grid components, sensors) whose behaviors are described by both the physical processes and the software layers characterizing them [15]. We started dealing with this problem in our previous work [9, 10], where we promote MAPE-K components as first-class modeling abstractions and provide an extensible framework supporting the design, development, and validation of decentralized self-adaptive cyber-physical systems. The framework provides a domain-specific environment for specifying the desired MAPE-K architecture and a cyber-physical simulation platform for simulating the designed self-adaptive system as a whole and producing simulation results. In particular, the CyPhEF tool[1] is implemented as an Eclipse plugin. The design level is based on the Eclipse Modeling Framework and the Graphical Modeling Project, which provide proper mechanisms for developing model-driven development (MDD) tools. On the other side, the simulation part leverages on co-simulation to simultaneously execute different models, allowing information to be shared among them.

The tool has been used to design and validate two different Smart Power Grid (SPG) real case studies. Specifically, SPG has been designed as a self-adaptive system that integrates the control system (i.e., managing subsystem) with an electrical power grid (i.e., managed subsystem) to produce and manage stable and sustainable electric energy.

4 FUTURE WORK

Developing future data-intensive systems satisfying dependability criteria is challenging and requires a radical shift towards the adoption of computing paradigms that fully reflect the dynamic and network-based nature of future execution environments. To this end, we adhere to the Edge Computing paradigm, which strives for the creation of scalable decentralized computing environments by pushing data and services away from central nodes to the edge of the network. Engineering such systems raises many challenges, as decentralization introduces uncertainty, which in turn may harm the dependability of the system itself. Therefore, we leverage self-adaptation techniques to mitigate uncertainties at runtime, and keep satisfying the requirements of the decentralized system.

Our research project is halfway through its five-year period. Hence, the vision and the contributions for the fulfillment of the research objectives are still to be considered partial. This paper summarizes some research results achieved in the first half of the research project.

For the second half of the research project, ongoing and future work proceed towards different lines of research. In particular, we want to deal with new real-world case studies with the aim of (*i*) evaluating the properties of different computing infrastructure supporting the data-intensive system domain, (*ii*) re-applying the techniques for uncertainties management that we have learned and, (*iii*) extending our framework to deal with these new scenarios. Specifically, future work aims at investigating **RQ1** and **RQ2** in the context of a real intelligent transport system scenario. In particular, we would like to evaluate decentralized vehicle-to-vehicle communication against centralized vehicle-to-cloud and vehicle-to-infrastructure management strategies. Moreover, regarding **RQ3**, we aim at extending our simulation framework by integrating an open-source microscopic traffic simulator [20].

[1]The interested reader can find the source code of the tool and demonstration videos at the following link: *https://mi-da.github.io/cyphef/*

REFERENCES

[1] D. P. Anderson. BOINC: A System for Public-Resource Computing and Storage. In *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, GRID '04, pages 4–10, Washington, DC, USA, 2004. IEEE Computer Society.

[2] F. Bonomi, R. Milito, J. Zhu, and S. 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, pages 13–16, New York, NY, USA, 2012. ACM.

[3] A. Bouguettaya et al. A Service Computing Manifesto: The Next 10 Years. *Commun. ACM*, 60(4):64–72, Mar. 2017.

[4] M. Caporuscio, M. D'Angelo, V. Grassi, and R. Mirandola. Reinforcement Learning Techniques for Decentralized Self-adaptive Service Assembly. 9846:289–298, 2016.

[5] V. Cardellini, M. D'Angelo, V. Grassi, M. Marzolla, and R. Mirandola. A Decentralized Approach to Network-Aware Service Composition. *Service Oriented and Cloud Computing*, 9306:V–VI, 2015.

[6] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021, 2017.

[7] V. D. Cunsolo, S. Distefano, A. Puliafito, and M. Scarpa. Volunteer Computing and Desktop Cloud: The Cloud@Home Paradigm. In *2009 Eighth IEEE International Symposium on Network Computing and Applications*, pages 134–139, July 2009.

[8] M. D'Angelo and M. Caporuscio. *Pure Edge Computing Platform for the Future Internet*, pages 458–469. Springer International Publishing, Cham, 2016.

[9] M. D'Angelo, M. Caporuscio, and A. Napolitano. Model-driven Engineering of Decentralized Control in Cyber-Physical Systems. In *2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, pages 7–12, 2017.

[10] M. D'Angelo, A. Napolitano, and M. Caporuscio. CyPhEF: A Model-Driven Engineering Framework for Self-Adaptive Cyber-Physical Systems. To appear at the 40th International Conference on Software Engineering (ICSE). Demonstrations Track. 2018.

[11] N. Esfahani and S. Malek. *Uncertainty in Self-Adaptive Software Systems*, pages 214–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[12] M. Galster and D. Weyns. Empirical Research in Software Architecture: How far have we come? In *2016 13TH Working IEEE/IFIP Conference on Software Architecture (WICSA) :*, pages 11–20, 2016.

[13] P. Mayer, A. Klarl, R. Hennicker, M. Puviani, F. Tiezzi, R. Pugliese, J. Keznikl, and T. Bure. The Autonomic Cloud: A Vision of Voluntary, Peer-2-Peer Cloud Computing. In *2013 IEEE 7th International Conference on Self-Adaptation and Self-Organizing Systems Workshops*, pages 89–94, Sept 2013.

[14] A. Montresor and M. Jelasity. PeerSim: A scalable P2P simulator. In *2009 IEEE Ninth International Conference on Peer-to-Peer Computing*, pages 99–100, Sept 2009.

[15] H. Muccini, M. Sharaf, and D. Weyns. Self-Adaptation for Cyber-Physical Systems: A Systematic Literature Review. In *2016 IEEE/ACM 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pages 75–81, May 2016.

[16] D. Perez-Palacin and R. Mirandola. Uncertainties in the Modeling of Self-adaptive Systems: A Taxonomy and an Example of Availability Evaluation. In *Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering*, ICPE '14, pages 3–14, New York, NY, USA, 2014. ACM.

[17] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8(4):14–23, Oct 2009.

[18] S. Sebastio, M. Amoretti, and A. Lluch Lafuente. A Computational Field Framework for Collaborative Task Execution in Volunteer Clouds. In *Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS 2014, pages 105–114, New York, NY, USA, 2014. ACM.

[19] K. Skala, D. Davidovic, E. Afgan, I. Sovic, and Z. Sojat. Scalable Distributed Computing Hierarchy: Cloud, Fog and Dew Computing. *Open Journal of Cloud Computing*, 2(1):16–24, 2015.

[20] M. Treiber and A. Kesting. An open-source microscopic traffic simulator. *IEEE Intelligent Transportation Systems Magazine*, 2010.

[21] A. u. R. Khan, M. Othman, S. A. Madani, and S. U. Khan. A survey of mobile cloud computing application models. *IEEE Communications Surveys Tutorials*, 16(1):393–413, First 2014.

[22] D. Weyns et al. On patterns for decentralized control in self-adaptive systems. *Lecture Notes in Computer Science*, 7475 LNCS:76–107, 2013.

[23] B. Xiao, R. Rahmani, Y. Li, T. Kanter, and D. Gillblad. Intelligent Data-Intensive loT: A Survey. pages 2362–2368, 2016.

[24] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young. Mobile Edge Computing A key technology towards 5G, 2015.