Towards a Fluid Cloud: An Extension of the Cloud into the Local Network

Bart Spinnewyn(✉) and Steven Latré

Department of Mathematics and Computer Science, University of Antwerp - iMinds, Middelheimlaan 1, 2020, Antwerp, Belgium
bart.spinnewyn@uantwerpen.be

Abstract. Cloud computing offers an attractive platform to provide resources on-demand, but currently fails to meet the corresponding latency requirements for a wide range of Internet of Things (IoT) applications. In recent years efforts have been made to distribute the cloud closer to the user environment, but they were typically limited to the fixed network infrastructure as current cloud management algorithms cannot cope with the unpredictable nature of wireless networks. This fixed deployment of clouds often does not suffice as, for some applications, the access network itself already introduces an intolerable delay in response time. Therefore we propose to extend the cloud formalism into the (wireless) IoT environment itself, incorporating the infrastructure that is already present. Given the mobile nature of local infrastructure, we refer to this as a fluid extension to the cloud, or more simply as a fluid cloud.

1 Introduction

While the Internet was originally intended as a dumb packet forwarder between fixed (large) computers, it now comprises a myriad of devices connected through a plethora of technologies and used for a wide range of applications. This unexpected evolution has fuelled the popularity and reach of the current Internet but has also put a significant strain on its management complexity. During recent years, two major trends have presented themselves that will define the Internet's management for the next decades: cloud computing and the Internet of Things. Cloud computing is enabled by large-scale datacentres hosting hundreds of thousands of servers, offering resources on a pay-per-use basis. The management of fixed cloud infrastructure has already extensively been studied in literature. Moens et al. propose a feature model, incorporating interdependencies between application features [1]. Zhani et al. advocate Virtual Datacentre Embedding (VDE) as a means of guaranteeing both server and network resources [2]. This is closely related to the work on Virtual Network Embedding (VNE) [3], [4], [5]. Here, virtual network components are embedded into a physical network substrate, similar to how virtual machines are embedded into a physical machine substrate. Another trend in the management of cloud computing is the inter-cloud paradigm, where not one cloud is considered but researchers focus on the optimal distribution of computational tasks between multiple clouds [6]. This

S. Latré et al. (Eds.): AIMS 2015, LNCS 9122, pp. 61–65, 2015.
DOI: 10.1007/978-3-319-20034-7_7

network of clouds is needed to ensure that some tasks are deployed on an infrastructure that is closer to the user. In more complex environments, a complete collection of private clouds is considered. Latency and jitter are becoming the dominant concerns in the application placement problem as they are typically used for applications that require a rapid response [7]. Nearby clouds generally perform better in terms of latency. This consideration eventually leads to the conception of fog computing. Urgaonkar et al. define the fog as a cloud close to the ground [7]. They propose placement of computational entities at the Internet Service Provider (ISP) level to lower latency and jitter.

A second important trend in communication networks is the recent emergence of the Internet of Things (IoT) paradigm. The developments in wireless technology and the price drop for miniaturized electronic components have fuelled the uptake of connected objects. Early products are in the meantime present in all market segments, and range from our personal devices and appliances over public infrastructures to industrial machinery. Future highly interactive applications will require ultra-low response times, these applications include robotics [8] and human-machine interactions [9]. Rapid growth of connected devices introduces new challenges e.g. with respect to ubiquitous wireless connectivity, efficient allocation of networking and computing resources, the real-time processing of continuous IoT data streams and the novel types of interactions that may emerge between humans and their smart, connected environment [10].

As a single sensor node typically does not have appropriate computational capabilities, tasks in an IoT environment are currently offloaded to a nearby server. While concepts such as cloud computing allow delegating these tasks to a computational infrastructure, it may not always be possible to use this centralized infrastructure e.g. because the introduced latency between the wireless link and fixed infrastructure is too high. Concepts such as the inter-cloud and fog computing are thus a first step towards solving this latency challenge but they are not able to support the required really fast response times for highly interactive applications.

2 Fluid Cloud

This paper proposes to extend the cloud formalism into the (wireless) IoT environment itself, incorporating the infrastructure that is already present and can be accessed at minimal latency. Given the mobile nature of the local infrastructure, we will refer to this as a fluid extension to the cloud, or more simply as a fluid cloud. This extension however is non-trivial as local infrastructure drastically differs from traditional cloud infrastructure. State-of-the-art management concepts assume infrastructure to be static, always-on, highly performing in terms of memory and CPU, and interconnected by an over-provisioned, well-controlled network. These assumptions clearly are no longer valid in the context of an IoT environment. Fluid cloud management must consider a broad spectrum, which includes devices that are low-power, unreliable and constrained in terms of connectivity, memory and processing. In the following we describe how we plan

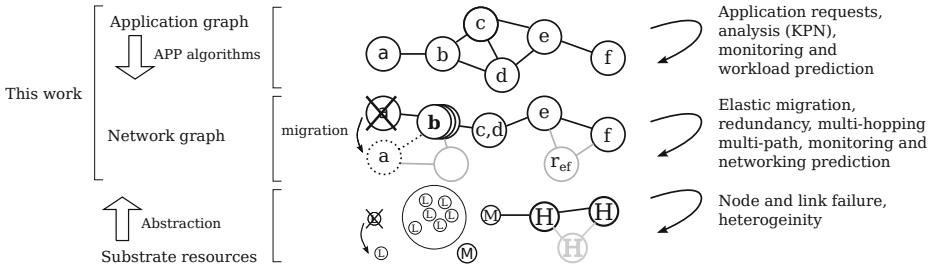

Fig. 1. Illustration of fluid cloud concepts: Processes c, d are consolidated onto one physical node, b is replicated onto multiple physical nodes. r_{ef} is a relay process that enables multi-hopping. A, b, c, d and c, d, e, f communicate over respectively wireless and wired infrastructure. Physical devices are non-restrictively labelled as either Low (L), Medium (M) or High (H) performance.

to overcome the challenges associated with extension of the cloud into an IoT environment. A high-level overview of the research project is given in Figure 1.

In a first phase, we will develop an IoT-aware application placement framework that handles requests for computational tasks and distributes these requests to available IoT infrastructure. Both initial and additional requests must be handled and the often dynamic QoS (Quality of Service) evolution in an IoT environment must be taken into account. Maximisation of the number of applications that can be hosted on the platform while satisfying resource limitations is referred to as the Application Placement Problem (APP), which in its general form is NP-hard [7], [11]. Efficient placement requires knowledge about the composition of applications. Therefore we will first develop a theoretical framework to describe applications as composed of processes, communicating over channels. We prefer a description in terms of processes as it allows use of elegant models such as Kahn Processing Networks (KPN) [12] and its extensions [13].

Secondly, we will use this theoretical framework for building new IoT-aware placement algorithms. Traditional application placement considers a list of machines, each having limited CPU and memory resources available and a list containing application processes, each having a certain demand for said resources. Limitations on machine resources, such as CPU and memory, will be incorporated as Knapsack constraints. Recent works incorporate bandwidth limitations as a third resource type. However such an approach implicitly assumes the networking interfaces of individual machines bottleneck throughput. This assumption is clearly not valid in an IoT environment, as the networking infrastructure cannot be assumed reliable and over-provisioned. Therefore, we will add topology-awareness to both application and networking description. In this context, we will extend the theoretical framework by translating application specifications into an application graph, displaying task-level parallelism and dependencies. Moreover, a network graph will be generated, representing Knapsack and QoS constraints as vertices and edges respectively. The network graph can be generated by instantiating monitoring components throughout the network.

Thirdly, traditional placement algorithms are said to be *binary*, indicating that a process is either placed once, or not at all. In an IoT environment, hosts are unreliable, as they can enter and leave the network at any time. We will mitigate this unreliable nature of hosts by grouped replication of processes. Individual processes can be placed multiple times, implicating that in our case the APP will be *non-binary*. Networking links can be highly unreliable. Therefore, on the other hand we will instantiate additional relay processes, enabling multi-hopping, a technique widely deployed in wireless mesh networks. This way placement algorithms gain control over routing, introducing an additional variable in QoS optimisation. Also we note that in QoS constrained applications, reliability of pair-wise communication between processes can be augmented by employing multiple parallel connections, referred to as multi-path routing.

In a second phase we will develop elastic migration algorithms to support migration of processes in live systems. Migration is called for when either networking environment or workload are altered significantly. In a QoS constrained system, migration tends to result in unacceptable delays, as this is a bandwidth and memory intensive procedure. Therefore, we will incorporate predictions to estimate future workload and networking conditions. Also, in our framework we will define multiple levels of QoS guarantees, each corresponding to certain networking conditions. As such we can define techniques of graceful degradation to cope with faults through system reconfiguration at runtime, at the cost of redundant active or standby resources. In this light, one needs to ensure the correct tasks restart after a system reconfiguration. Therefore, we will implement check-pointing protocols to support state reservation.

Acknowledgments. Part of this work has been funded by the iFEST and EMD project, co-funded by iMinds and IWT.

References

1. Moens, H., Truyen, E., Walraven, S., Joosen, W., Dhoedt, B., De Turck, F.: Feature placement algorithms for high-variability applications in cloud environments. In: 2012 IEEE Network Operations and Management Symposium (NOMS), pp. 17–24. IEEE (2012)
2. Zhani, M.F., Zhang, Q., Simon, G., Boutaba, R.: VDC planner: Dynamic migration-aware virtual data center embedding for clouds. In: 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), pp. 18–25 (2013)
3. Fischer, A., Botero, J.F., Till Beck, M., De Meer, H., Hesselbach, X.: Virtual network embedding: A survey. IEEE Communications Surveys & Tutorials 15(4), 1888–1906 (2013)
4. Mijumbi, R., Gorricho, J.-L., Serrat, J., Claeys, M., Turck, F.D., Latré, S.: Design and evaluation of learning algorithms for dynamic resource management in virtual networks. In: 2014 IEEE Network Operations and Management Symposium (NOMS), pp. 1–9. IEEE (2014)

5. Latre, S., Famaey, J., De Turck, F., Demeester, P.: The fluid internet: service-centric management of a virtualized future internet. IEEE Communications Magazine 52(1), 140–148 (2014)
6. Buyya, R., Ranjan, R., Calheiros, R.N.: InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services. In: Hsu, C.-H., Yang, L.T., Park, J.H., Yeo, S.-S. (eds.) ICA3PP 2010, Part I. LNCS, vol. 6081, pp. 13–31. Springer, Heidelberg (2010)
7. Urgaonkar, B., Rosenberg, A.L., Shenoy, P.: Application placement on a cluster of servers. International Journal of Foundations of Computer Science 18(05), 1023–1041 (2007)
8. Hu, G., Tay, W.P., Wen, Y.: Cloud robotics: architecture, challenges and applications. IEEE Network 26(3), 21–28 (2012)
9. Wang, A.L., Canedo: Offloading industrial human-machine interaction tasks to mobile devices and the cloud. In: 2014 IEEE Emerging Technology and Factory Automation (ETFA), pp. 1–4 (September 2014)
10. Essa, I.A.: Ubiquitous sensing for smart and aware environments. IEEE Personal Communications 7(5), 47–49 (2000)
11. Camati, R.S., Calsavara, A., Lima Jr., L.: Solving the virtual machine placement problem as a multiple multidimensional knapsack problem. In: The Thirteenth International Conference on Networks, ICN 2014, pp. 253–260 (2014)
12. Vrba, Z., Halvorsen, C., Beskow, P.: Kahn process networks are a flexible alternative to MapReduce. In: 11th IEEE International Conference on High Performance Computing and Communications, HPCC 2009, pp. 154–162. IEEE (2009)
13. Copil, G., Moldovan, D., Truong, H.-L., Dustdar, S.: Sybl: An extensible language for controlling elasticity in cloud applications. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 112–119 (May 2013)