



**HAL**  
open science

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

Bart Spinnewyn, Steven Latré

► **To cite this version:**

Bart Spinnewyn, Steven Latré. Towards a Fluid Cloud: An Extension of the Cloud into the Local Network. 9th Autonomous Infrastructure, Management, and Security (AIMS), Jun 2015, Ghent, Belgium. pp.61-65, 10.1007/978-3-319-20034-7\_7. hal-01410148

**HAL Id: hal-01410148**

**<https://hal.science/hal-01410148>**

Submitted on 6 Dec 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Towards a Fluid Cloud: an Extension of the Cloud into the Local Network

Bart Spinnewyn, Steven Latré

University of Antwerp, iMinds, Department of Mathematics & Computer Science,  
Middelheimlaan 1, 2020 Antwerp, Belgium

**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

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.

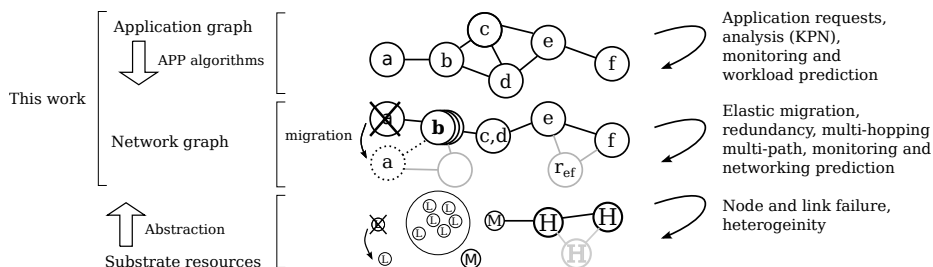
A second important trend in communication networks is the recent emergence of the Internet of Things (IoT) paradigm. The developments in wireless technology and the price drop for miniaturized electronic components have fuelled the uptake of connected objects. Early products are in the meantime present 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

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.



**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.

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.

### 3 Conclusion

Cloud computing initially focused on placing computational power at the end of the network. In recent years however, we clearly observe the reverse trend: computational power is more and more being integrated with the network itself. With a few small-scale exceptions, the computational capacity however remains in the fixed network and is not integrated into the wireless network itself. This paper proposes a fluid extension to the cloud, besides achieving lower latency in task execution, such a fluid cloud can facilitate hand-overs and can even execute applications without the need of a fixed infrastructure. Integrating the cloud in a wireless network is however non-trivial and highly challenging as the environments are much more chaotic and unreliable. There is thus a clear need for more fundamental cloud management algorithms that can support the unreliable nature of wireless environments.

### Acknowledgments

The authors would like to thank IWT (Flemish agency for Innovation by Science and Technology) as this work is partially funded by ICON projects iFEST and EMD.

## References

1. Hendrik Moens, Eddy Truyen, Stefan Walraven, Wouter Joosen, Bart Dhoedt, and Filip De Turck. Feature placement algorithms for high-variability applications in cloud environments. In *Network Operations and Management Symposium (NOMS), 2012 IEEE*, pages 17–24. IEEE, 2012.
2. M.F. Zhani, Qi Zhang, G. Simon, and R. Boutaba. VDC planner: Dynamic migration-aware virtual data center embedding for clouds. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 18–25, 2013.
3. Andreas Fischer, Juan Felipe Botero, M Till Beck, Hermann De Meer, and Xavier Hesselbach. Virtual network embedding: A survey. *Communications Surveys & Tutorials, IEEE*, 15(4):1888–1906, 2013.
4. Rashid Mijumbi, Juan-Luis Gorricho, Joan Serrat, Maxim Claeys, Filip De Turck, and Steven Latré. Design and evaluation of learning algorithms for dynamic resource management in virtual networks. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–9. IEEE, 2014.
5. S. Latre, J. Famaey, F. De Turck, and P. Demeester. The fluid internet: service-centric management of a virtualized future internet. *Communications Magazine, IEEE*, 52(1):140–148, January 2014.
6. Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N Calheiros. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In *Algorithms and architectures for parallel processing*, pages 13–31. Springer, 2010.
7. Bhuvan Uргаonkar, Arnold L Rosenberg, and Prashant Shenoy. Application placement on a cluster of servers. *International Journal of Foundations of Computer Science*, 18(05):1023–1041, 2007.
8. Guoqiang Hu, Wee Peng Tay, and Yonggang Wen. Cloud robotics: architecture, challenges and applications. *Network, IEEE*, 26(3):21–28, May 2012.
9. Lingyun Wang and A. Canedo. Offloading industrial human-machine interaction tasks to mobile devices and the cloud. In *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*, pages 1–4, Sept 2014.
10. I.A. Essa. Ubiquitous sensing for smart and aware environments. *Personal Communications, IEEE*, 7(5):47–49, Oct 2000.
11. Ricardo Stegh Camati, Alcides Calsavara, and Luiz Lima Jr. Solving the virtual machine placement problem as a multiple multidimensional knapsack problem. In *ICN 2014, The Thirteenth International Conference on Networks*, pages 253–260, 2014.
12. Zeljko Vrba, Carsten Halvorsen, and Paul Beskow. Kahn process networks are a flexible alternative to MapReduce. In *High Performance Computing and Communications, 2009. HPCC'09. 11th IEEE International Conference on*, pages 154–162. IEEE, 2009.
13. G. Copil, D. Moldovan, Hong-Linh Truong, and S. Dustdar. Sybl: An extensible language for controlling elasticity in cloud applications. In *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pages 112–119, May 2013.