Microcities: a Platform based on Microclouds for Neighborhood Services

Ismael Cuadrado-Cordero, Felix Cuadrado, Chris Phillips, Anne-Cécile Orgerie, Christine Morin

Ismael Cuadrado-Cordero, Felix Cuadrado, Chris Phillips, Anne-Cécile Orgerie, Christine Morin. Microcities: a Platform based on Microclouds for Neighborhood Services. [Research Report] RR-8885, inria. 2016. <hal-01291536>

HAL Id: hal-01291536
https://hal.inria.fr/hal-01291536
Submitted on 21 Mar 2016

Microcities: a Platform based on Microclouds for Neighborhood Services

Ismael Cuadrado-Cordero, Felix Cuadrado, Chris Phillips,
Anne-Cécile Orgerie, Christine Morin

ISRN INRIA/RR--8885--FR+ENG

SN 0249-6399

RESEARCH
REPORT
N° 8885
Fevrier 2016

Project-Teams Myriads,
Networks (QMUL)

Microcities: a Platform based on Microclouds for Neighborhood Services

Ismael Cuadrado-Cordero, Felix Cuadrado, Chris Phillips,
Anne-Cécile Orgerie, Christine Morin

Project-Teams Myriads, Networks (QMUL)

Research Report n° 8885 — Fevrier 2016 — 18 pages

Abstract: The current mobile Cloud computing trend has set the focus on the ubiquity of computation. However, the current architecture confines the cloud to datacenters, which are generally far from the user. Distance leads to increased utilization of the broadband Wide Area Network - WAN - and poor user experience, especially for interactive applications. Decentralized architectures are emerging as an alternative, but they still fail to adapt to situations where files are concurrently modified. A semi-decentralized approach which confines local traffic close to the user while still maintaining centralized characteristics, running on the users and network devices, can provide a better Quality of Experience (QoE) in large urban populations in mobile cloud networks. In this report, we propose a novel semi-centralized cloud architecture based on microclouds. Microclouds are dynamically created and allow users to contribute resources from their computers, mobile and network devices to the cloud. We present the process for building and reconfiguring the microclouds, whilst guaranteeing a high QoE to users of real-time applications. We also provide a description of a realistic mobile cloud use-case. Results from a simulation-based evaluation indicate that the microclouds architecture is able to sustain hundreds of mobile devices and provide a latency significantly lower than regular Clouds based on datacenters.

Key-words: Cloud Computing, Internet-of-Things, Mobile Cloud, Networking

CENTRE DE RECHERCHE

RENNES - BRETAGNE-ATLANTIQUE

Microcities: a Platform based on Microclouds for Neighborhood Services

Résumé : La tendance actuelle des applications mobiles dans les nuages informatiques (clouds) a mis láccent sur lómniprésence du calcul. Cependant, lárchitecture actuelle confine le cloud dans les centres de données qui sont généralement loin de lútilisateur. Cette distance conduit á une utilisation accrue de la bande passante des réseaux longue distance (Wide Area Network - WAN) et conduit á une expérience utilisateur médiocre, en particulier pour les applications interactives. Les architectures décentralisées apparaissent comme une alternative, mais elles ne parviennent pas toujours á s'adapter á des situations oú les fichiers sont modifiés de maniére concurrente. Une approche semi-décentralisée, mise en oeuvre sur les équipements des utilisateurs et du réseau et confinant le trafic local á proximité de lútilisateur tout en conservant les caractéristiques centralisées, peut fournir une meilleure qualité déxpérience (QoE) aux grandes populations urbaines utilisant les applications mobiles des clouds. Dans cet rapport, nous proposons une nouvelle architecture de cloud semi-centralisée fondée sur les "microclouds". Les microclouds sont créés dynamiquement et permettent aux utilisateurs de fournir au cloud des ressources de leurs ordinateurs, smartphones et équipements réseau. Nous présentons le processus de construction et de reconfiguration des microclouds, tout en garantissant un haut degré de QoE aux utilisateurs dápplications temps-réel. Nous donnons également une description dún cas d'utilisation réaliste de cloud mobile. Les résultats dúne évaluation menée par simulation indiquent que lárchitecture á base de microclouds est en mesure de gérer des centaines d'équipements mobiles et de fournir un temps de latence nettement inférieur á celui des clouds traditionnels fondés sur les centres de données.

Mots-clés : Informatique en nuage, Internet des objets, informatique mobile, réseau

1 Introduction

The wide uptake of Cloud architectures have caused global IP traffic to increase five fold [8], catalyzed by the ubiquity of mobile devices. According to Cisco, global IP traffic is envisioned to increase threefold over the next five years, with mobile wireless traffic exceeding wired traffic by 2016 [8]. This adds to the limitations of mobile devices in terms of resources and connectivity [10], bringing new challenges to the Cloud. In order to address these issues, a new paradigm is emerging: Mobile Cloud computing. This paradigm improves resource-hungry mobile services such as Internet data sharing [19], wearables [3] or augmented reality [18], by offloading data and computation into the Cloud [10].

This offloading of computation requires high-speed connectivity between the clients and the datacenter. However, prevailing highly geographically-centralized Cloud architectures do not properly handle it. Fernando et al. argue that considering data access fees, latency, bandwidth and energy consumption of wireless connectivity, a Mobile local Cloud - constrained to the location of the user - is a better alternative than a remote one [10]. In addition, local Clouds also provide locality-awareness and support for latency-critical interactions, thus providing a better Quality-of-Experience.

In this report, we go a step further into Mobile Cloud computing by integrating the mobile devices themselves into the Cloud architecture. Our proposed architecture, called microclouds, introduces a flexible, semi-decentralized and yet efficient solution for locality-related applications. We build a local cloud on top of networking resources and the mobile devices of users spread across a defined area. This local cloud is managed by lightweight mechanisms which handle the dynamicity of users who can appear/disappear and move.

Our contributions are: 1) a semi-decentralized mobile cloud architecture called microclouds, 2) a description of a realistic mobile cloud use-case in line with smart cities, and 3) a simulation-based evaluation of our architecture. Our results indicate that the microclouds architecture is able to sustain one hundred mobile devices and provide a latency significantly lower than regular Clouds based on large datacenters.

The remainder of this report is structured as follows. Section 2 provides the context and motivation and we establish the case study of this work. In Section 3, we define the design and implementation of our system. In Section 4 we analyze experimentation results. Finally, Section 5 highlights our key findings and draws directions to future work.

2 Motivation and Scenario

2.1 Context and Motivation

The use of centralized datacenters is, nowadays, the most realistic approach regarding the deployment of heavy computation services. This architecture relies on a robust communication infrastructure between distant clients, obtaining a computing power otherwise unattainable. However, centralized architectures suffer downsides such as traffic delays, replication of data and scalability issues related with the physical constraints of datacenter resources. Moreover, it forces other actors involved in the communication (such as Internet Service Providers) to oversize their infrastructure in consequence [6].

As studied in [24], [13] and [7], information propagation in real-life is usually not distant. This is because interactions in the Internet are conditioned by our interactions as a society. Groups of interest are generally geographically constrained. This is the case with sports teams (whose fans are generally located in the same area), departments in a company (whose users are located in the same building) or Geographic Information Systems [17].

This situation is, by design, approached by Cloud computing, as it is described as a versatile and ubiquitous system. However, in reality, Cloud platforms run on large centralized datacenters, which provide the needed infrastructure. Yet, since existing infrastructures cannot effectively host ever increasing demands, datacenters need to be expanded, which is costly and requires advanced planning. In fact, this situation has already been anticipated by host providers, which have started distributing their datacenters around the globe, balancing the flow of information.

The use of centralized systems in users' inclusive - citizens are both providers and consumers of information - and non-heavy computation services with low propagation is, thus, inappropriate. A distributed approach, where the flow of information is not only produced and consumed, but also processed in the same area is, in this case, desirable. The main advantages of distributed approaches in this scenario are low latency, scalability, and adaptability. This kind of approach fits the mobile clouds created in smart cities initiatives all around the world (Santa Cruz, Amsterdam, Barcelona, etc.) where local authorities deploy wireless platforms to manage traffic or emergency situations.

Even if the proportion of geographically constrained traffic is difficult to estimate, traffic characterization in different areas can be found in literature. For example, in [15] the authors evaluate the consumed and generated traffic in a rural African village. In this scenario, the authors show that most of the generated and consumed traffic is of a local scope, with web and social networking services being the most utilized ones. In addition, in [1] the authors characterize usage of a freely available outdoor wireless Internet in California (USA). Their results show a peak of smartphone connections in transit areas, while in residential areas the connections are more balanced between static and mobile. Commercial areas show a higher activity than either transit or residential areas.

In this report, we propose a semi-decentralized Cloud architecture for localized communities such as neighborhoods. We propose a mobile Cloud case study based on smart cities initiatives and provide a practical study and evaluation combining mobile and static devices - other existing Cloud infrastructures provide either a mobile peer-to-peer network [19] or local clouds with connectivity to remote cloud servers [16]. We show that, through the use of microclouds, latency is greatly reduced - compared to centralized systems - while providing a robust, elastic and adaptive platform of services. Traffic in datacenters, and network providers' broadband utilization and transit costs are also reduced.

2.2 Related Work

Two main approaches in the literature are of interest to the case study.

Centralized Clouds are based on a specific infrastructure to which all clients connect. An issue commonly attributed to these architectures is the excessive distance between clients and the computation infrastructure. Adaptability is also a problem as once the infrastructure has been allocated, it is expensive and complicated to extend. To provide a service ”closer to the user” that is better able to handle the increase in demand of computation, Cloud providers - such as Google [12] or Amazon [2] - disperse their datacenters around the globe.

Centralized local clouds reduce the distance between the user and the infrastructure. In [23], the authors describe an environment where mobile devices outsource their computation to a Cloudlet - domestic servers which provide cloud services to a relatively small set of users. It reduces the computational load of the devices, the replication of data, and the delay, and provides a service adapted to users' needs. However, they require a, sometimes, prohibitive investment, they are rigid and the dependence on the infrastructure ties the user to the system.

Distributed Clouds are not deployed on the same infrastructure but among several independent nodes, providing a more robust and adaptable system compared to a lone infrastructure that offers a single point of failure. A thorough review of distributed technologies is shown in [4]. One of the more praised benefits of distributed approaches is the low latency experienced by users. Representative examples of distributed cloud systems are Content Distribution Networks (CDNs) [5] and Peer-To-Peer (P2P) Clouds [22], [26] - systems where individual computers are distributed across the globe and connected through the Internet hosting VMs for parallel computation, content distribution or storage.

Decentralized approaches also have disadvantages, such as network restrictions (distributed approaches are rather network demanding) and excessive replication of content. As a solution, the union of distributed and centralized architectures (semi-centralized) is also covered in literature. In [25], authors propose the replication of data in the Internet Provider's datacenter in order to reduce latency and energy consumption. However, it still suffers from higher delay than a totally decentralized approach, in addition to an excessive replication (for example, in between users connecting to different providers).

2.3 Scenario: Neighborhood Services

Neighborhood-related applications are a good example of geographically localized services. In a neighborhood, many services are only of interest to the community, like street works, water or electricity cuts or local store information (stocks, opening hours, etc.). These networks are heterogeneous - comprising both mobile and fixed nodes provided by both citizens and city infrastructures that serve up and consume information [14]. Social networks such as [20] and [11] - where users share and interact information of interest only with their neighbors - are examples of neighborhood-oriented applications. A system appropriate for this environment should adjust to the following characteristics:

Non-Replication: Data can be classified in *Announcements*, which are immutable and distributed through the network to inform of an event; and *collaborative/interactive information* (shared documents, forums, etc.), which have several contributors and are prone to version conflicts and extensive replication in non-centralized systems (broadband utilization in multipurpose networks is very dynamic, and information may flood the network if not handled correctly). Therefore, no replication of data and/or management is allowed.

Adaptability, **Virtualization** and **Splitting:** Due to a *dynamic workload*, *different communication patterns for different neighborhoods* and *variability in investment for different neighborhoods*, the platform needs to dynamically adapt its topology. Also, virtualization and the ability to split services into users groups allows migration and robustness and enhances the *isolation* and *coexistence of services*, providing different services for different neighborhoods.

Network Orientation, **Locality Awareness** and **Disruptive Behavior:** Since a neighborhood exists within a relatively small geographical area, the system should take into account the physical location of the nodes to obtain the best possible utilization of the network. Knowing the location of the nodes, the platform is able to manage *information dissemination*, adapting to the *available bandwidth*. Due to the interactive nature of *real-time information*, disruptive behavior (asynchronous communication) is not desirable.

3 Our Solution: Microcities

3.1 Architecture Design

Our platform is semi-decentralized and extends the concept of microcloud proposed in [9] in order to support dynamic interactivity of multiple services with no replication. A microcloud is an overlay network that connects independent users working using the same service - for each service a microcloud is created. Each microcloud hosts a Light Virtual Machine or LVM - a type of operating-system-level virtualization which runs the service. Using the LVM, data are centralized for each service, while microclouds distribute the computation across the network. All roles and processes described below are transparent to the user.

Base Node (BN): Stores LVMs - while no client is using them - and security policies. This logical role is taken by a server assumed to be highly available, and provisioned by a public authority in charge of the neighborhood.

Manager: Two types of manager exist:

- **Base Manager (BM):** This role is also taken by a resilient and trustworthy device - it can be the same as for the base node - which ensures an inefficiency threshold in the system. It periodically evaluates the QoE of the system and if it exceeds a given threshold - set by the administrator and adapted to the specific characteristics of the infrastructure - requests a new topology and/or an LVM migration to one or more microclouds.

- **Service Manager (SM):** Controls the topology of the microcloud and distributes the roles (service provider and linking nodes). This role is assigned by the base manager depending on the nodes' computing capacity and robustness. Only one service manager exists per microcloud, and it is preferably assigned to a stable, static node.

Service Provider (SP): Hosts the LVM(s). This role is assigned by the service manager based on the minimum delay between the service provider and every client, reliability in time and hardware capabilities, and it is unique in the microcloud. It is also preferably assigned to a stable, static node.

Linking Node: Links to nodes belonging to the same or different microclouds (noted in Figure 1 using a double line). If a linking node retransmits data to and from a node it is referred to a **Bridge**. Otherwise, if it forwards non-interactive information from the *SP* to one or more clients, a **Repeater**.

Client: Consumer of information, that is, a user. A physical node can take several client roles (one per microcloud), isolated from each other, or even different roles (such as client, service provider and/or service manager). Before using the service, it follows the joining process described in Section 3.2.

To improve the robustness of the system, the roles of SM and SP are replicated across different nodes, called backup nodes, following a hierarchical structure transparent to the user. If either the manager or the service provider is confirmed to have failed, through several reports sent to the *BM* from nodes in the system which cannot contact it, their role is taken by the next node in the hierarchy. If the dead node is running the SP role, the LVM is relaunched in the node chosen as a backup (to do so, the nodes selected as backups periodically retrieve a snapshot of the LVM). The LVM is migrated if the node acting as a service provider is excluded from the microcloud after the BM requests a reconfiguration. Figure 1 shows the logical distribution of nodes in our use-case.

3.2 Join and Detach Processes

The connection process is depicted in Figure 2 and described below. It is launched by the user and extends the DEEPACC protocol presented in [9] to find the fastest path between the client and the *SP*. All services are listed in the *BM*, and the user selects one of them through a graphical interface. Failures in communications and reconfigurations triggered by the *BM* due to excessive inefficiency cause a reconnection. The process is the same, but is not initiated without a user request.

1. The client obtains the address of the *SP* through a request to the base manager. After that, a connection request is sent towards the *SP*, flooding the network through every possible path. In every node, the message is captured, processed and updated with the current Round-Trip Time (RTT).

2. Once the *SP* obtains all the possible paths, they are sent to the service manager. The *SM* uses a Branch and Bound algorithm [9] to plan the microcloudś topology and communicates this topology to the *SP* and to the *BM*. The *BM* uses this information for inefficiency-triggered reconfigurations.

3. The *SP* sends an acceptance message to the client with its address and a list of the nodes in the route through the chosen route. Before forwarding the message, every node that intercepts it, updates its routing table and extracts itself from the list until the client receives the message.

Disconnections/failures follow the process depicted in Figure 3. This is launched once a user disconnects from the service (properly or due to a failure in communications). This disconnection is either processed or discovered and communicated to the *SM*, which restructures the network to keep providing the service to the remaining nodes.

Figure 2 shows the process of joining a microcloud. Figure 3 shows the process of properly detaching from a microcloud. If a link is broken, all affected nodes relaunch the DEEPACC protocol, which restarts the joining process.

3.3 Dependability

Despite the unreliability of the environment, the design of our system provides a certain resilience to failures. On one hand, the existence of a base infrastructure

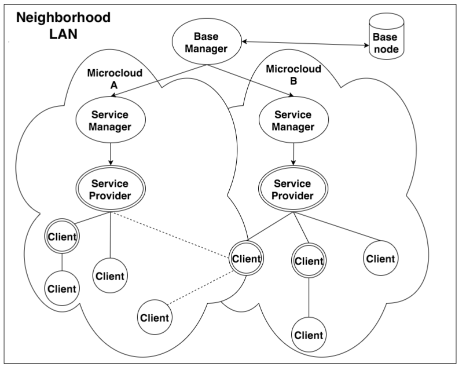

Figure 1: Neighborhood overlay services (microclouds)

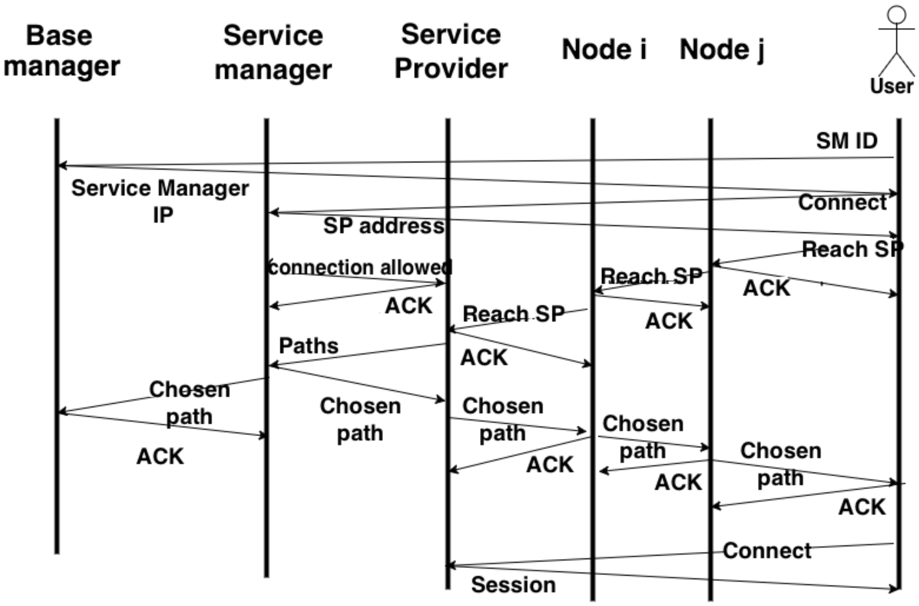

Figure 2: Joining a microcloud

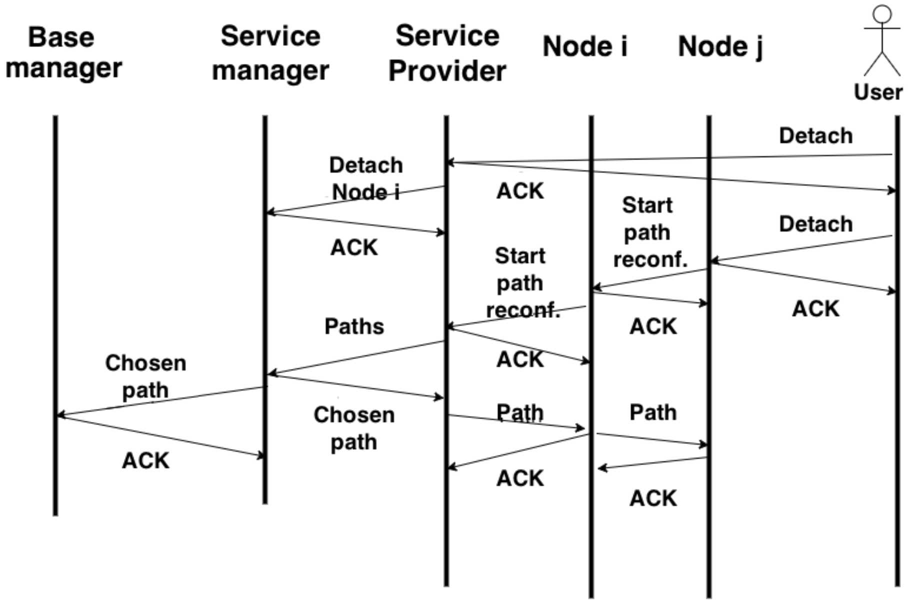

Figure 3: Detachment from a microcloud

Figure 4: Protocol description

ensures that, even when no clients make use of a LVM, a working copy of data always exists. On the other hand, during microcloudś runtime, data are backed up in one or more nodes in the same microcloud. If either the *SP* or the *SM* are determined to be down, a joining process is repeated using a backup node as the main provider or manager.

Moreover, a microcloud is aware of connectivity and/or physical failures through a keepalive protocol. That way, every client in a microcloud is responsible for the state of its connection with its main neighbor (the first hop between the node and the *SP*), running a periodic check. If a link dies, a new joining process is launched by the client which detected the failure. Every other participant connecting through this client will remain connected to it, to accelerate the reconnection of the microcloud. Finally, the same service can be provided by a large microcloud or several smaller ones (split from the original microcloud), which offers higher resilience - as a node crash would affect a smaller number of nodes if a large microcloud is split in several smaller ones.

4 Evaluation

The aim of our experiments is to evaluate the suitability of a microclouds platform for operation within a neighborhood, compared with a centralized approach. To do so, we focus on the user experience in terms of the average packet delay between nodes, the overhead of the routing protocol (extra communication time), packet loss probability and the reliability of the system. Due to scalability issues, we extend the results of a 10 nodes prototype using a simulation on NS3 [21]. The simulated network contains a variable number of mobile nodes - with a random mobility in the plane - over a physical network of 45 static nodes.

As a first evaluation, we compare the communication delay perceived by the user, shown in Figure 5. It measures the Round-Trip Time (RTT) both in microclouds and in a centralized cloud approach. For the centralized approach, Amazon EC2 has been chosen due to its representativeness, and we launch the experiments in different availability zones to provide different RTT values. A 45-minute real trace of a shared on-line document session has been used. Due to scale of the network, the results shown have been simulated in an environment comprising 110 nodes (small neighborhood). Data fed to the simulation have been obtained from a 10-client prototype.

Results are explained by the distance between users in the use-case (almost negligible compared to Amazon's world-wide area). As shown, microclouds provide an average delay of about 15 ms, while the centralized experiment shows a RTT several times higher. It is also worth noting that we have not observed packet loss using either microclouds or EC2. Additionally, Figure 6 shows the RTT evolution when the number of nodes in the microcloud increases. As expected, the delay depends on the number of nodes and the distance in between.

Second, we evaluate the robustness of our approach compared to a cen-

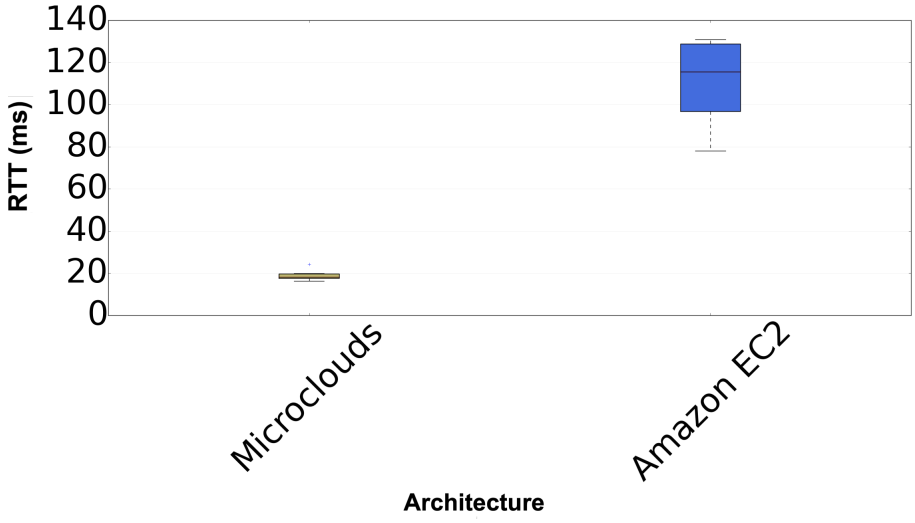

Figure 5: microclouds vs Amazon EC2 RTT

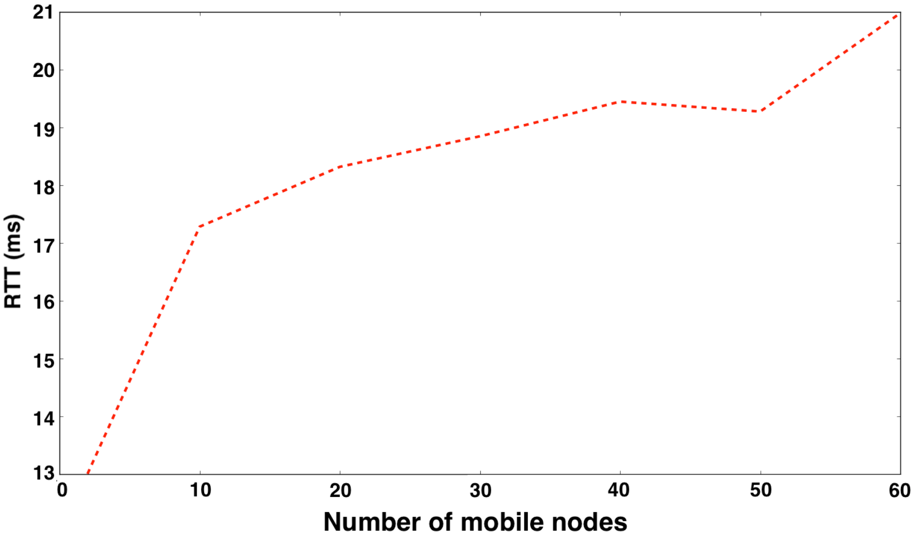

Figure 6: Total microclouds RTT in % of time

Figure 7: Microclouds RTT comparison

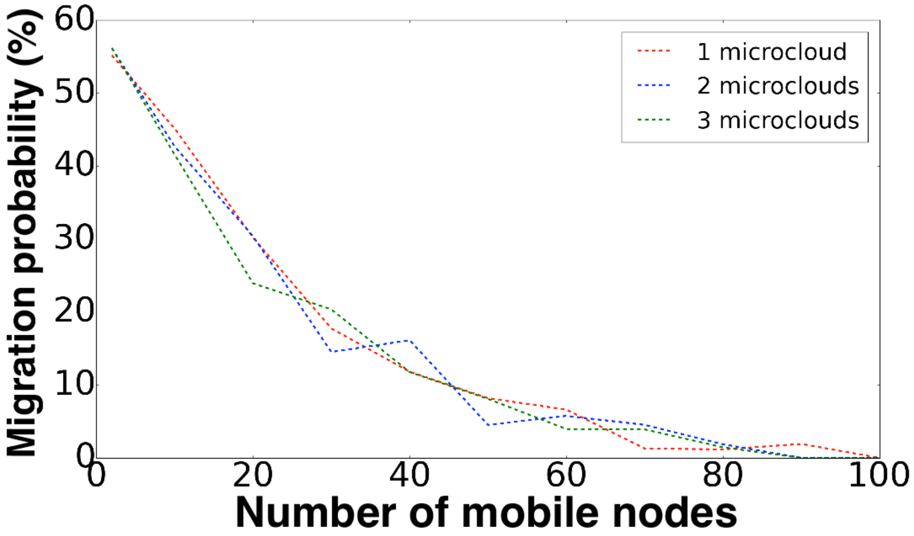

Figure 8: Probability of migration of LVM

tralized approach. We assume no infrastructure failures (which affects both approaches equally) and evaluate the probability of service interruptions caused by LVM migrations. An LVM may be migrated after a reconfiguration. Although the migration process adds a very small delay to the overall computation time, frequent migrations are not desirable.

Figure 8 shows the probability of migrating a LVM using either one big or several small microclouds' configurations, compared with the number of clients in the network, shown on Figure 9. For small microclouds (10 or fewer clients), the probability of the LVM being migrated is around 50%. It rapidly decays with the addition of new nodes as the microcloudś topology converges to that of the physical one until all the nodes participate in the overlay network and so no migration is needed. To avoid excessive migration, an inefficiency heuristic is needed to reduce the migrations whilst providing a better QoE than centralized approaches, adapted to the network capabilities based on experimentation.

Migration of LVMs produce, likewise, extra computation due to the joining process and calculation of new routes. While the service is not halted until a new path is computed and established, excessive extra computation affects the QoE and the demands the platform makes of the finite resources. To avoid excessive computational overhead, an inefficiency threshold may be set, so no reconfiguration is launched below this threshold. For our experimentation, we used a heuristic where no reconfiguration is launched until, at least, 50% of the nodes involved in the original overlay network change.

Figure 13 compares the computational overhead of a threshold-triggered redistribution and a reconnection triggered redistribution (launched when a mobile

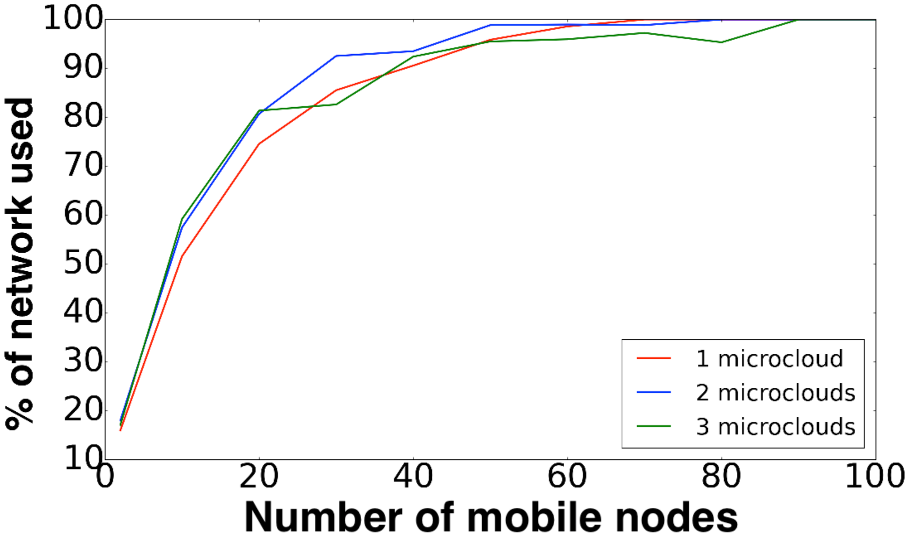

Figure 9: Utilization of the network

Figure 10: Probability of migration of a LVM vs. utilization of the network

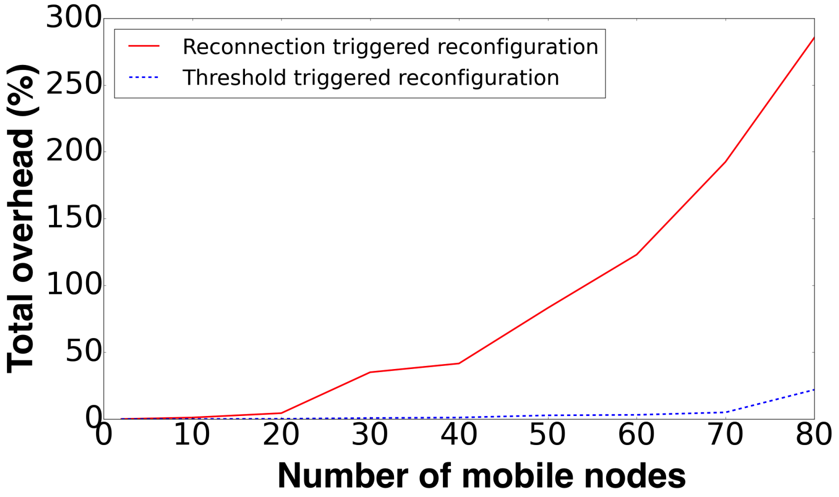

Figure 11: Total overhead

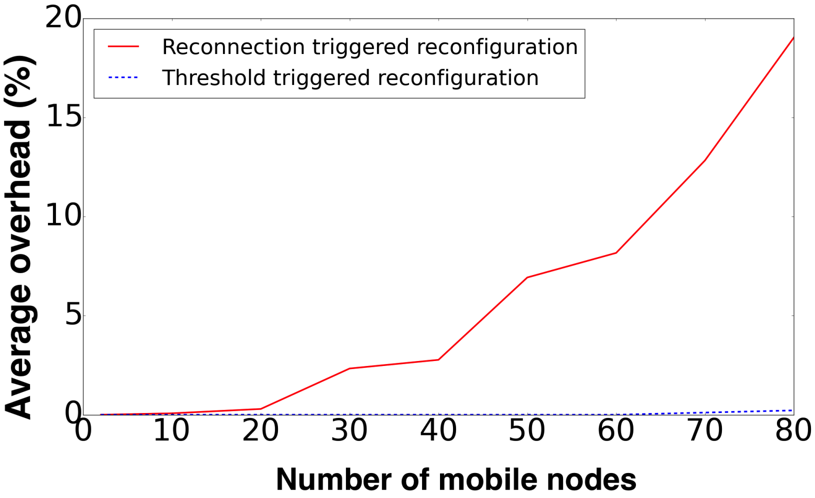

Figure 12: Average overhead

Figure 13: Overhead comparison (in % of time)

node connects through a new node). As shown on Figure 11, the total computation time added by the joining process exceeds almost 3 times the one used for the remainder of the of the 45-minute trace, while using a threshold triggered approach it remains under 25% of this time. On average - compared to the total execution time -, as shown in Figure 12, the use of a threshold triggered approach is almost imperceptible compared to the rest of the computation time.

Finally, we evaluate the overhead in time that our approach adds due to the computation of the routing path compared with a centralized one for the node acting as the SM. This overhead has been calculated on a MacBook Pro, with a 2.7 GHz Intel Core i7 and 8 GB 1600 MHz DDR3, representing a *SM*. This overhead is also reduces by splitting the microcloud into smaller ones - thus increasing decentralization but reducing the route-planning computation. Figure 16 shows the total and average overhead along the duration of the trace.

We can see that the overhead increases exponentially and is directly related to the number of clients in the microcloud - which affects the size of the network and the number of reconnections of nodes, which causes network planning. As shown, when the network size is around 100 of mobile nodes, the total overhead reaches a 40% of the time. However, due to the ability to split microclouds, this time can be easily reduced by distributing the computation among as many microclouds as necessary. Since the path computation is distributed among different nodes, the average overhead computation on each SM is reduced. Given that each microcloud manages fewer nodes, the total overhead is also reduced.

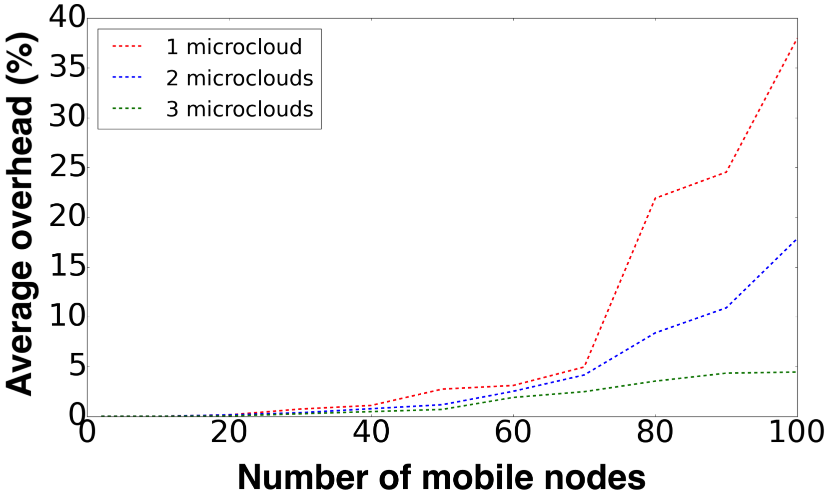

Figure 14: Total overhead

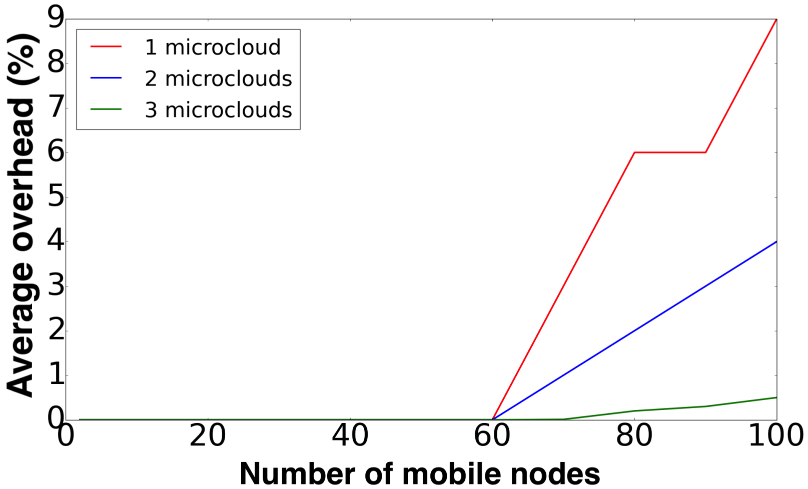

Figure 15: Average overhead

Figure 16: Overhead comparison splitting microclouds (in % of time)

5 Conclusions and Future Work

In this report we evaluated the suitability of a microcloud-based platform - a semi-distributed approach to managing geographically constrained data - in the context of mobile clouds. We investigated a neighborhood service-based scenario. Given that microclouds' design is semi-distributed and network-oriented, our approach exploits network resources to reduce unnecessary data movement over long distance networks. We evaluated the RTT, overhead time and robustness of our approach compared to a totally centralized approach. We show that, in the studied scenario, our platform provides better QoE in terms of latency and utilization of the networking resources - in terms of bandwidth utilization and profit from network infrastructure - over the centralized approach, whilst still providing a robust service. Also, we propose the use of an inefficiency-based heuristic to ensure the adaptability of the system to the network, and to reduce the need for extra-computation to deal with network planning.

We are now working on a pricing model for microclouds, as an extra incentive to adopt a microclouds-based system. The exchange of computational time for monetary revenue would boost the uptake of microclouds. Service creators would benefit from the open market and the increase in competition, while clients would receive a faster and more personalized service, together with financial benefits from making their computational resources available for hosting or relaying service information. Together with this, we plan to work on mobile congestion issues and refine our protocol, as appropriate, and address possible security issues associated with microclouds.

References

[1] M. Afanasyev, Tsuwei Chen, G.M. Voelker, and A.C. Snoeren. Usage Patterns in an Urban WiFi Network. *IEEE/ACM Trans. on Networking*, 18(5):1359–1372, 2010.

[2] AmazonTM. Amazon cloud infrastructure.

[3] O. Amft and P. Lukowicz. From Backpacks to Smartphones: Past, Present, and Future of Wearable Computers. *IEEE Pervasive Computing*, 8(3):8–13, July 2009.

[4] O. Babaoglu and M. Marzolla. The people's cloud. *IEEE Spectrum*, 51(10):50–55, October 2014.

[5] Spiridon Bakiras and Thanasis Loukopoulos. Combining replica placement and caching techniques in content distribution networks. *Computer Communications*, 28(9):1062 – 1073, 2005.

[6] M. Bertier et al. Beyond the Clouds: How Should Next Generation Utility Computing Infrastructures Be Designed? In Zaigham Mahmood, editor, *Cloud Computing*, Computer Communications and Networks, pages 325–345. Springer International Publishing, 2014.

[7] Meeyoung Cha, Alan Mislove, and Krishna P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *International Conference on World Wide Web (WWW)*, pages 721–730, 2009.

[8] Cisco. The zettabyte era: Trends and analysis. Technical report, 2015.

[9] Ismael Cuadrado Cordero, Anne-Cécile Orgerie, and Christine A. Morin. GRaNADA: A Network-Aware and Energy-Efficient PaaS Cloud Architecture. In *IEEE International Conference on Green Computing and Communications (GreenCom)*, December 2015.

[10] Niroshinie Fernando, Seng W. Loke, and Wenny Rahayu. Mobile cloud computing: A survey. *Future Generation Computer Systems*, 29(1):84 – 106, 2013.

[11] Goneighbour, 2010.

[12] GoogleTM. Google cloud.

[13] Desislava Hristova, Mirco Musolesi, and Cecilia Mascolo. Keep your friends close and your facebook friends closer: A multiplex network approach to the analysis of offline and online social ties. *CoRR*, abs/1403.8034, 2014.

[14] IEC. Orchestrating infrastructure for sustainable Smart Cities. Technical report, International Electrotechnical Commission, 2015.

[15] D. L. Johnson, Elizabeth M. Belding, and Gertjan van Stam. Network traffic locality in a rural african village. In *International Conference on Information and Communication Technologies and Development (ICTD)*, pages 268–277, 2012.

[16] M.D. Kristensen. Scavenger: Transparent development of efficient cyber foraging applications. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 217–226, March 2010.

[17] Paul A Longley, Michael F Goodchild, David J Maguire, and David W Rhind. *Geographic information science and systems.* John Wiley & Sons, 2015.

[18] Xun Luo. From Augmented Reality to Augmented Computing: A Look at Cloud-Mobile Convergence. In *IEEE International Symposium on Ubiquitous Virtual Reality (ISURV)*, pages 29–32, July 2009.

[19] E. Marinelli. Hyrax: Cloud Computing on Mobile Devices using MapReduce. Master's thesis, Carnegie Mellon University, 2009.

[20] Nextdoor, 2010.

[21] Ns3 simulator. `"https://www.nsnam.org/"`, 2015.

[22] Zhao Peng, Huang Ting-lei, Liu Cai-xia, and Xin Wang. Research of P2P architecture based on cloud computing. In *Int. Conf. on Intelligent Computing and Integrated Systems (ICISS)*, pages 652–655, Oct 2010.

[23] Mahadev 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.

[24] Salvatore Scellato, Cecilia Mascolo, Mirco Musolesi, and Jon Crowcroft. Track globally, deliver locally: Improving content delivery networks by tracking geographic social cascades. In *International Conference on World Wide Web (WWW)*, pages 457–466, 2011.

[25] Vytautas Valancius, Nikolaos Laoutaris, Laurent Massoulié, Christophe Diot, and Pablo Rodriguez. Greening the internet with nano data centers. In *International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, pages 37–48, 2009.

[26] Ke Xu, Meina Song, Xiaoqi Zhang, and Junde Song. A Cloud Computing Platform Based on P2P. In *Int. Symp. on IT in Medicine Education (ITIME)*, pages 427–432, 2009.

CENTRE DE RECHERCHE

RENNES - BRETAGNE-ATLANTIQUE

Éditeur
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr
ISSN 0249-6399