Towards Dynamic Fog Resource Provisioning for Smart City Applications

José Santos*, Tim Wauters*, Bruno Volckaert* and Filip De Turck*
* Ghent University - imec, IDLab, Department of Information Technology
Technologiepark-Zwijnaarde 15, 9052 Gent, Belgium
Email: josepedro.pereiradossantos@UGent.be

***Abstract*— Over the past few years, the Internet of Things (IoT) has transformed the Smart City concept into an attractive and relevant opportunity. Smart Cities aim to connect billions of objects of everyday life to the Internet to improve sustainability and citizen welfare. Fog Computing has been introduced to provide scalable and low latency services for the future IoT use cases by placing cloud resources at the edges of the network. Nevertheless, crucial challenges still remain in the Fog Computing domain. One of them is providing proper resource provisioning while reducing allocation costs, maximizing energy efficiency and minimizing latency. Therefore, in this paper, a novel Resource Discovery Service based on Peer-to-Peer (P2P) Distributed Hash Tables (DHT) is presented to enable automated resource discovery functionalities for IoT services. The proposed approach provides a flexible way of exchanging resource allocation information between the Fog and the Cloud Layer. Evaluations have been conducted to demonstrate the performance of implementing information dissemination systems based on DHTs. Performance ratios higher than 99%, including latencies under 0.7 seconds for a static context have been obtained. Results show that DHTs provide scalable discovery solutions showing the full applicability of the proposed approach in the Smart City environment.**

***Index Terms*—Fog Computing, Resource Provisioning, Smart Cities, IoT, P2P, DHT**

I. Introduction

In recent years, Smart City [1] has become a popular term for making smart use of Internet of Things (IoT) devices to transform public services in different domains of urban life, such as environmental monitoring, public transportation and emergency services. The Fog Computing paradigm [2] has been introduced by placing cloud resources on the edges of the network close to end devices, thus helping to mitigate the demanding constraints introduced by IoT, such as high mobility and low latency. The so-called Fog Nodes [3] are small cloud entities distributed across the network to allocate and deploy services in a decentralized manner. However, these deployment procedures imply proper resource provisioning, which is not a simple operation. Resource allocation issues are present in different domains of literature related to the life-cycle management of applications and services. This is still a key research domain in Fog Computing [4]. One of the challenges is how to set up and maintain the infrastructure of each individual Fog Node while dealing with all the service requests received from the IoT devices. Furthermore, another challenge that still remains is how to enable proper allocation functionalities for IoT services. Multiple factors should be taken into account to ensure a proper resource provisioning such as response time, bandwidth usage, allocation cost and energy efficiency. Nevertheless, few resource provisioning strategies are currently addressing the service placement issues in Smart Cities imposed by the demanding constraints of IoT.

In this paper, a novel Resource Discovery Service based on Peer-to-Peer (P2P) Distributed Hash Tables (DHTs) is proposed to deal with the service placement issues in Smart Cities. Nodes are organized in a distributed manner without any hierarchy or centralized control in a DHT. The main operation in a DHT is the information lookup, where a node requests a data value based on a unique Identifier (ID) called a key. The main advantage of DHTs is the logarithmic performance in lookup operations since requests are satisfied in a bounded number of steps even for large-scale distributed architectures. DHTs provide a flexible way of exchanging provisioning information between Fog and Cloud Nodes, such as the free amount of computing resources on a certain Node (e.g. RAM available, free Disk Space) and the service level information of a particular service instance (e.g. allocated amount of bandwidth). The proposed approach enables automated resource discovery functionalities leading to a dynamic and decentralized resource provisioning in Smart Cities. Evaluations have been carried out to demonstrate the performance of implementing information dissemination systems based on DHTs. Several DHTs have been assessed to find the most appropriate protocol to be applied in a Smart City context.

The remainder of this paper is structured as follows. In the next section, related work is discussed. Section III introduces the proposed Resource Discovery Service. Then, in Section IV, the evaluation approach is described which is followed by the results in Section V. Finally, conclusions are presented in section VI.

II. Related Work

In recent years, research efforts have been carried out to deal with service placement issues in Smart Cities. In [5], a self-configurable P2P architecture for service discovery in IoT is presented. In the proposed solution, the IoT gateway acts as a backbone, which keeps track of any device joining or leaving the network. Furthermore, in [6], a DHT-based Discovery Service for IoT is presented, however, the inherent requirements of IoT applications, especially distributed operations, low latency and high mobility are not addressed. Moreover, a

978-3-903176-14-0 © 2018 IFIP

federated discovery service named ForwarDS-IoT is discussed in [7]. The proposal aims to semantically describe resources and IoT services to provide efficient discovery functionalities. Nevertheless, Fog Computing concepts are not addressed in the cited proposals, which limit the applicability of their approach, specifically for delay-sensitive and high mobility use cases since centralized management approaches cannot fully satisfy the latency and dynamicity demands of these type of applications.

This work builds further on [8], where a resource provisioning Integer Linear Programming (ILP) formulation for Low Power Wide Area Networks (LPWAN) in Smart Cities has been presented. The ILP model takes into account not only cloud requirements but also wireless constraints while optimizing multiple demanding IoT requirements, such as latency, service migrations and energy efficiency. In this paper, a novel Resource Discovery Service is proposed since the objective of our approach is not only focused on identifying services allocated on the network and keeping track of them as in most ongoing research projects, but also providing an efficient way of exchanging provisioning information between Fog and Cloud Nodes to rapidly identify available computing resources that can be used to allocate and instantiate Smart City services. For instance, exchanging the amount of computing resources available on Fog and Cloud Nodes periodically allows for more informed resource allocation decisions, as up-to-date information is accessible on the state of the computational resources. To the best of our knowledge, no suitable way to provide this resource information dissemination is currently available in the literature. By combining P2P decentralized solutions and Fog Computing architectures, the proposed approach paves the way towards dynamic resource provisioning in the Smart City ecosystem.

III. A Resource Discovery Service based on DHTs

Fig. 1 presents the proposed Resource Discovery Service for Smart City applications based on DHTs. Architectural components of the Fog Node and Cloud Node are also illustrated. As shown, Fog and Cloud Nodes are distributed across the network to provide resources to end devices and users. Each Fog Node is an autonomous system managing a limited set of resources. The Cloud Nodes are the top level management entities, which are responsible for the global control of the network. Further explanations on the envisioned functionalities for the Fog Node and the Cloud Node entities are discussed in [10]. Additionally, a DHT is established between Fog and Cloud Nodes. DHTs form logical topologies built on top of physical networks to store key and value pairs without any hierarchy or centralized control. The main operation in a DHT is called lookup, where a peer requests a data value based on a unique ID named as key. Lookup operations allow nodes to retrieve (*GET*) information from the overlay. When a lookup operation is issued, the request is routed through the overlay until it reaches the peer responsible for the requested lookup(key). On the other hand, when the information is stored in the overlay, it is called a *PUT* request. Several DHT protocols have been proposed in the literature over the years, among the most popular are Chord [11], Pastry [12] and Kademlia [13].

As a first step in the design of the Resource Discovery Service, the provisioning information needed to be stored in the DHT is determined. Each service available in the network possesses Service Templates which are stored in the Service Inventory module. The Service Templates are the containers of all the information needed to onboard, register and instantiate services in Fog and Cloud Nodes. The Service Templates are inspired from the OpenBaton [9] VNF information model. OpenBaton is an ETSI Network Function Virtualization (NFV) compliant Orchestrator (NFVO). Fig. 2 provides an example of a Service Template. When a service instance is provisioned by a Fog or a Cloud Node, the resource operational characteristics of the service instance are stored in the DHT. The key of the record is based on the hash of the service instance, which is unique through the network. By storing this information, it is possible to know exactly which computing resources have been allocated to a particular instance because all the needed provisioning information is available through the DHT. Furthermore, the static and dynamic configurations of each Fog and Cloud Node are also stored in the DHT. The static configurations are related to the total computational resources available on a certain Node (amount of CPUs, Disk space and RAM Memory). Additionally, it is also important to accurately store what a particular Node can offer at a certain moment in terms of idle CPUs, megabytes of RAM and gigabytes of Disk space. These attributes are related to the dynamic configuration. These records must be constantly updated in the DHT so that the dynamic configurations can be known at every moment by the other Nodes. Therefore, the Time To Live (TTL) of these records is rather low due to the volatile information. Thus, each Node knows the amount of computing resources allocated for a certain service instance and has an approximate view on the amount of resources still available in the network for the provisioning of services.

In summary, their logarithmic performance and their decentralized nature make DHTs an effective solution for exchanging resource operational information in Fog Computing architectures.

IV. Evaluation Approach

A. Evaluation Method

Multiple simulators have been developed to analyze P2P systems over the past years. In [14], a comprehensive survey of the available P2P simulators in literature is presented. Based on their work, the OverSim [15] simulator has been selected for the evaluation due to its flexibility, modular design and high scalability. In the evaluation, data items are stored and distributed within the DHT. Then, lookup operations are generated from random peers for all the data items originally stored. Bandwidth and jitter parameters are assigned to nodes based on the simple underlay network model available in OverSim. TCP and UDP constant delay are set to 50ms and jitter to 0.1.

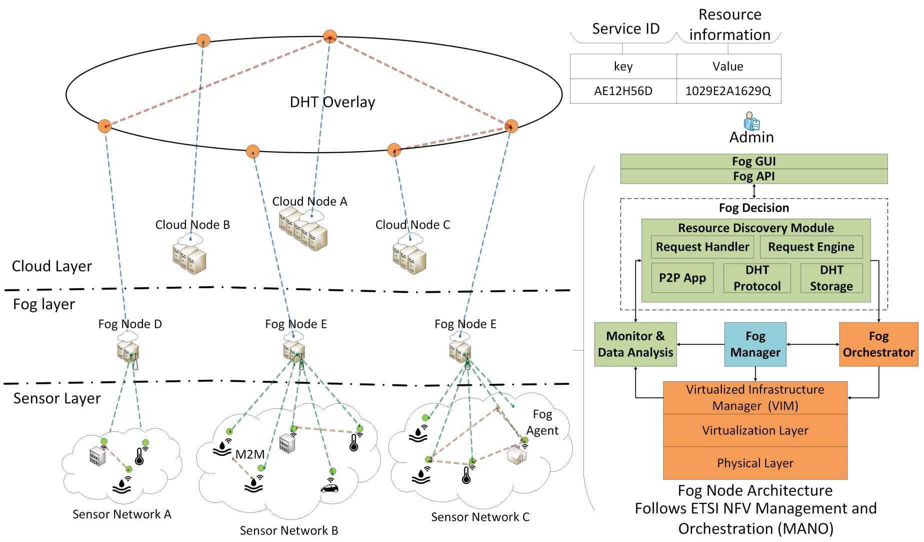
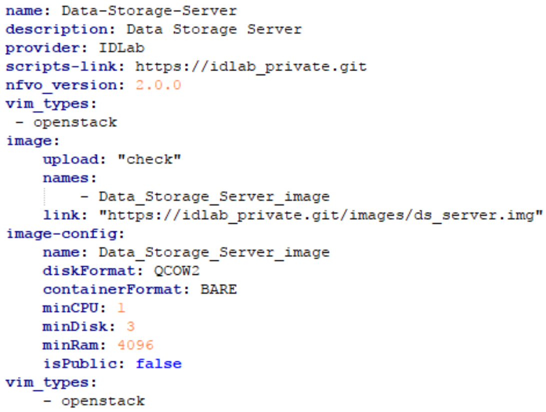


Fig. 1: Overview of the proposed Resource Discovery Service for Fog Computing architectures in Smart Cities.



```
name: Data-Storage-Server
description: Data Storage Server
provider: IDLab
scripts-link: https://idlab_private.git
nfvo_version: 2.0.0
vim_types:
 - openstack
image:
    upload: "check"
    names:
        - Data_Storage_Server_image
    link: "https://idlab_private.git/images/ds_server.img"
image-config:
    name: Data_Storage_Server_image
    diskFormat: QCOW2
    containerFormat: BARE
    minCPU: 1
    minDisk: 3
    minRam: 4096
    isPublic: false
vim_types:
    - openstack
```

Fig. 2: An example of a Service Template [9].

B. Evaluation Scenarios

Priority services, such as emergency applications and fire rescue services must be rapidly reallocated when a failure happens on the Node allocating this type of services or when the Quality of Service (QoS) levels surpass the thresholds (e.g. High Latency). For instance, the vehicles are moving across the City and constantly sending requests for specific service instances. At a given moment, latency may increase since vehicles are constantly moving in the City. Therefore, service migrations are needed to reduce the latency. By adopting the proposed approach, Fog or Cloud Nodes can obtain the provisioning requirements of those particular service instances through the DHT and then apply fast and smooth service migrations by allocating the necessary resources to keep the provisioned services operating properly without latency. Two scenarios have been assessed, a static and a dynamic context. In the static scenario, Fog and Cloud Nodes are considered to be always connected to the overlay during the simulation. However, malfunctions and failures may happen on a given Node leading to a leave operation, which is commonly known as *churn*. Both scenarios have been evaluated 30 times and confidence intervals of 95% have been considered.

V. Evaluation Results

A. Static Scenario

In this assessment, random PUT and GET requests have been generated in 5 second intervals, iterative routing has been selected, i.e. the generated lookup is sent to the closest next hop rather than the entire DHT, thus reducing the bandwidth usage. Furthermore, the TTL of each record has been set up to 120 seconds since the provisioning information needed to store in the DHT is quite volatile. Additionally, the number of peers has been set to 10 and 100, because it has been assumed that a relatively low and high Fog Computing deployment in a Smart City context could require in the order of 10 and 100 Fog Nodes respectively. Finally, the Replication Factor (RF) has been set up to 1 (no replication), meaning that there is only one record stored per key-value pair. The other simulation parameters assigned to each DHT protocol where exactly the same so that each protocol has been assessed in the same situation. The obtained results for 1 hour of simulation time are presented in Table I and in Table II for a low and high Node deployment respectively. As shown, Pastry requires on average 100ms less time than Kademlia and 400ms less than Chord to answer GET requests for a high Fog Computing deployment (100 Nodes). Furthermore, both Kademlia and Chord successfully answer GET requests faster for a low Fog Computing deployment (10 Nodes). The reduction in latency for Pastry is not so evident for a low Node deployment. The achieved results for the dynamic scenario are presented next.

B. Dynamic Scenario

In this assessment, nodes are affected by churn. The churn is performed in a two-step process as explained in [16]. The

TABLE I: DHT Evaluation: Static Scenario (10 Nodes)

DHTs	**Performance Metrics**			
	GET Latency (s)	GET Ratio (%)	Normal (Bytes/s)	Maintenance (Bytes/s)
Chord	0.65	99.27	118.53	3.1×10^{-3}
Kademlia	0.41	99.58	118.61	0
Pastry	0.51	99.44	118.59	4.8×10^{-3}

TABLE II: DHT Evaluation: Static Scenario (100 Nodes)

DHTs	**Performance Metrics**			
	GET Latency (s)	GET Ratio (%)	Normal (Bytes/s)	Maintenance (Bytes/s)
Chord	0.93	99.13	118.63	6.8×10^{-3}
Kademlia	0.64	99.32	118.64	0
Pastry	0.54	99.42	118.66	5×10^{-3}

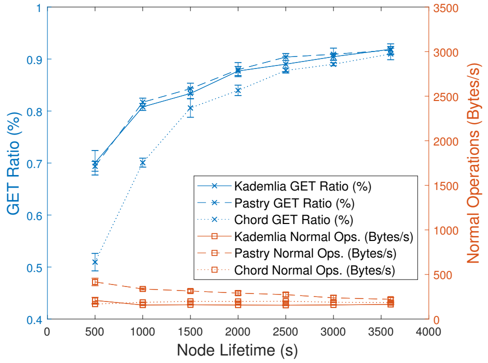


Fig. 3: DHT: GET Ratio & Normal Ops. for RF 2.

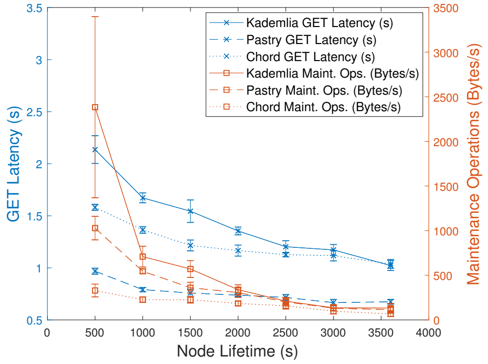


Fig. 4: DHT: GET Latency & Maint. Ops. for RF 2.

deadtimeMean parameter has been set to 300 seconds (5 minutes) while the *lifetimeMean* has been increasing throughout the multiple simulations. The *lifetimeMean* parameter represents the average time the node is connected. Then, the node is removed based on the *deadtimeMean* parameter, which represents the mean time that the node is disconnected. The high Node deployment has been considered and the other simulation parameters have been kept equal to the ones used in the static scenario. However, the RF is increased to 2 (two records peer each DHT entry). Fig. 3 shows the evolution of the percentage of successful GET requests and the bandwidth usage for Normal operations while Fig. 4 shows the average latency in responding to GET requests and the bandwidth usage for Maintenance operations. As expected, churn affects the reliability and performance. Kademlia and Pastry achieve practically the same GET ratio while Chord achieves a slightly lower ratio (6% less). Furthermore, Pastry responds to GET requests on average 0.4 seconds faster than Kademlia and Chord when churn is not so evident (Node Lifetime higher than 2000 seconds), however, the bandwidth usage to obtain it is quite high. Kademlia requires only 62% of the needed bandwidth of Pastry for normal operations. However, regarding maintenance operations, Kademlia requires on average 7% and 32% more bandwidth than Pastry and Chord respectively due to the increase of replication.

In summary, Chord is unfeasible for our approach because it takes barely less than one second to respond successfully to GET requests for a high Node deployment in a static context. Furthermore, Pastry has a clear trade-off between response time and bandwidth. Pastry is slightly faster than Kademlia, however, it requires a high amount of bandwidth. In our opinion, Kademlia is the most suitable protocol for our approach because achieves good performance ratios and low latencies with a reasonable use of bandwidth even for a highly dynamic scenario.

VI. Conclusions

This paper presents a novel Resource Discovery Service based on DHTs, which provides a scalable manner of exchanging resource allocation information in Fog Computing architectures. The proposed approach extends the current state-of-the-art within Fog Computing by enabling autonomous resource discovery functionalities for the provisioning of IoT services in Smart Cities. Evaluations have been conducted to demonstrate the feasibility of developing information dissemination systems based on DHTs. Results show the effectiveness of using DHTs for the exchange of provisioning information. The evaluation demonstrated that both Kademlia and Pastry are appropriate candidates for the proposed approach. Performance ratios higher than 99%, including latencies under 0.7 seconds for a static scenario have been achieved. Both protocols provide flexible discovery solutions showing the full applicability of our approach in the Smart City environment. As future work, proof-of-concepts of our approach will be implemented and validated through real service deployments in Antwerp's City of Things (CoT) testbed.

Acknowledgment

This research was performed partially within the projects ”Service-oriented management of a virtualised future internet” and ”Intelligent DEnse And Longe range IoT networks (IDEAL-IoT)” under Grant Agreement #S004017N, both from the fund for Scientific Research-Flanders (FWO-V). The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO and the Flemish Government – department EWI.

References

[1] V. Albino, U. Berardi, and R. M. Dangelico, “Smart cities: Definitions, dimensions, performance, and initiatives,” *Journal of Urban Technology*, vol. 22, no. 1, pp. 3–21, 2015.

[2] S. Yi, C. Li, and Q. Li, “A survey of fog computing: concepts, applications and issues,” in *Proceedings of the 2015 Workshop on Mobile Big Data*. ACM, 2015, pp. 37–42.

[3] S. Yi, Z. Hao, Z. Qin, and Q. Li, “Fog computing: Platform and applications,” in *Hot Topics in Web Systems and Technologies (HotWeb), 2015 Third IEEE Workshop on*. IEEE, 2015, pp. 73–78.

[4] Y. Jiang, Z. Huang, and D. H. Tsang, “Challenges and solutions in fog computing orchestration,” *IEEE Network*, 2017.

[5] S. Cirani, L. Davoli, G. Ferrari, R. Léone, P. Medagliani, M. Picone, and L. Veltri, “A scalable and self-configuring architecture for service discovery in the internet of things,” *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 508–521, 2014.

[6] F. Paganelli and D. Parlanti, “A dht-based discovery service for the internet of things,” *Journal of Computer Networks and Communications*, vol. 2012, 2012.

[7] P. Gomes, E. Cavalcante, T. Rodrigues, T. Batista, F. C. Delicato, and P. F. Pires, “A federated discovery service for the internet of things,” in *Proceedings of the 2nd Workshop on Middleware for Context-Aware Applications in the IoT*. ACM, 2015, pp. 25–30.

[8] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, “Resource provisioning for iot application services in smart cities,” in *CNSM 2017, the 13th International Conference on Network and Service Management*, 2017, pp. 1–9.

[9] (2017) Open baton: an open source network function virtualisation orchestrator (nfvo) fully compliant with the etsi nfv mano specification. [Online]. Available: http://openbaton.org

[10] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, “Fog computing: Enabling the management and orchestration of smart city applications in 5g networks,” *Entropy*, vol. 20, no. 1, p. 4, 2018.

[11] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, “Chord: a scalable peer-to-peer lookup protocol for internet applications,” *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 1, pp. 17–32, 2003.

[12] A. Rowstron and P. Druschel, “Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems,” in *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer, 2001, pp. 329–350.

[13] P. Maymounkov and D. Mazieres, “Kademlia: A peer-to-peer information system based on the xor metric,” in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 53–65.

[14] S. Surati, D. C. Jinwala, and S. Garg, “A survey of simulators for p2p overlay networks with a case study of the p2p tree overlay using an event-driven simulator,” *Engineering Science and Technology, an International Journal*, vol. 20, no. 2, pp. 705–720, 2017.

[15] I. Baumgart, B. Heep, and S. Krause, “OverSim: A flexible overlay network simulation framework,” in *Proceedings of 10th IEEE Global Internet Symposium (GI ’07) in conjunction with IEEE INFOCOM 2007, Anchorage, AK, USA*, May 2007, pp. 79–84.

[16] Z. Yao, D. Leonard, X. Wang, and D. Loguinov, “Modeling heterogeneous user churn and local resilience of unstructured p2p networks,” in *Network Protocols, 2006. ICNP’06. Proceedings of the 2006 14th IEEE International Conference on*. IEEE, 2006, pp. 32–41.