Distribution of Semantic Reasoning on the Edge of Internet of Things

Xiang Su, Pingjiang Li, Jukka Riekki, Xiaoli Liu
University of Oulu
Oulu, Finland

Jussi Kiljander, Juha-Pekka Soininen
VTT Technical Research Centre of Finland
Oulu, Finland

Christian Prehofer
fortiss, An-Institut Technische Universität München
Munich, Germany

Huber Flores
University of Helsinki
Helsinki, Finland

Yuhong Li
Beijing University of Posts and Telecommunications
Beijing, China

***Abstract*—Semantics associates meaning with Internet of Things (IoT) data and facilitates the development of intelligent IoT applications and services. However, the big volume of the data generated by IoT devices and resource limitations of these devices have given rise to challenges for applying semantic technologies. In this article, we present Cloud and edge based IoT architectures for semantic reasoning. We report three experiments that demonstrate how edge computing can facilitate IoT systems in terms of data transfer and semantic reasoning. We also analyze how distributing reasoning tasks between the Cloud and edge devices affects system performance.**

I. Introduction

Internet of Things (IoT) connects physical objects with sensing, networking, and processing capabilities to the Internet. These devices generate large amounts of data that needs to be represented, stored, searched, organized, and utilized.

Semantics associates meaning with data, thus allowing interpretation of data in context. For IoT, semantic technologies encode meaning into IoT data to enable computer systems to possess knowledge and support decision making. Semantic technologies based on machine-interpretable representations facilitate sharing and integrating IoT data, modelling and querying information, and infering new knowledge. For example, semantic sensor web [1] enables annotating IoT data with spatial, temporal, and thematic semantic metadata to create situation awareness. However, semantic representations and reasoning techniques require a considerable amount of resources. The volume of data generated by IoT devices and resource limitations of these devices have given rise to challenges for applying semantic technologies in IoT systems.

Processing IoT data requires deploying intelligent functions at different components of IoT systems in order to support accurate, comprehensive, and timely decision making and actions. Moreover, the data should be represented in a way that heterogeneous and resource-limited IoT devices can understand and utilize it in a convenient way. Cisco suggests requirements for IoT systems, including minimization of data processing latency, conservation of bandwidth consumption, collecting and securing data across wide geographic areas, and addressing security, privacy, and system reliability concerns [2].

Novel IoT architectures are needed for fulfilling these requirements and providing services with high performance and quality. Edge computing addresses these challenges with moving the computation from the central Cloud or server machines to the edges of IoT networks. The targeted benefits of edge computing result from its proximity to data sources and end users: 1) low and predictable latency for end users and applications; 2) secure and privacy-preserving services and applications; 3) long battery life and low bandwidth cost; and 4) scalability [3][4]. Edge computing balances the workload of IoT system components and improves user experience.It is predicted that 45% of IoT data will be stored, processed, analyzed, and acted upon close to, or at the edge of the network by 2019 [5].

This article focuses on semantic reasoning in IoT systems with edeg devices. We use Resource Description Framework (RDF) [6] as the semantic data model which thus provides an approach for heterogeneous machines to understand and utilize the data. We present an experimental IoT system that has semantic reasoners both on Cloud and edge devices for performing reasoning tasks. We design three experiments to demonstrate how edge computing could facilitate IoT systems in terms of data transferring and semantic reasoning. We also analyze how to improve the performance by distributing reasoning tasks on the Cloud and edge devices.

The main contribution of this article is a comparison and analysis of semantic reasoning in Cloud and edge computing based IoT architectures in a smart transportation use case. In the experimentation, we utilize real data collected from taxi drivers in Oulu, Finland. The reminder of this article is organized as follows: Section II presents background and related work. Section III describes the system design and scenarios. We introduce two architectures in Section IV and present experiments and analysis based on these architectures in Section V. Finally, we conclude the paper with suggesting future research in Section VI.

II. Background and Related Work

A. Edge and Fog Computing

The edge computing trend starts from Mobile Edge Computing (MEC) [7], which reduces the network workload by shifting computational efforts from the core network to the mobile edge. The first real-world MEC platform, called Radio

Applications Cloud Servers [8], was introduced by Nokia in 2014.

Fog computing pushes the processing capability further down to the data sources [2]. Data can be processed and stored either in fog computing nodes close to the data sources, in fog aggregation nodes, or in Cloud servers. Cisco Fog Computing Solutions [2] provide connectivity for a wide range of IoT devices, considering data security, data processing priorities, and automatic provision. When compared with edge computing, fog computing solutions often have less stringent constraints in terms of hardware and application execution model.

The most fundamental challenges for edge and fog computing are how to decompose, distribute, and compose computational tasks over a set of heterogeneous nodes with limited communication and computational capabilities [9]. Vögler et al. [10][11] present LEONORE infrastructure for provisioning IoT applications on fog computing nodes in large scale IoT systems. Giang et al. develop IoT applications that span across Cloud and fog computing nodes. Their distributed dataflow programming model specifies a generic methodology for distributing computation over Fog and Cloud computing nodes [12]. Abdelwahab et al. introduce Long-term Evolution (LTE)-aware Edge Cloud infrastructure and LTE-optimized memory optimization protocol for IoT applications [13]. Sharing a similar vision of utilizing edge and fog computing in IoT systems, our research focuses on distributing semantic data analytics on edge nodes of IoT systems.

B. Semantic Technologies

Semantic technologies facilitate data integration, resource discovery, system interoperability, semantic reasoning, and knowledge extraction for IoT systems [14]. To achieve these goals, IoT data needs to be represented in machine-interpretable formats [14][15]. World Wide Web Consortium (W3C) has developed a family of Semantic Web standards. The key technologies utilized in this research include RDF, RDF Schema (RDFS), Web Ontology Language (OWL), and semantic reasoners.

RDF is flexible in representing arbitrary structure without a *priori* schema. RDF uses a graph-based data model, where a graph consists of statements with (subject, predicate, object) structure. This structure can be interpreted as: "object o stands in relationship p with subject s". RDF can be represented in different serialization formats including RDF/XML [16], JSON for Linked Data (JSON-LD) [17], N-Triples [18], N-Quads [19], Turtle [20], RDFa [21], Notation 3 (N3)[22] and Entity Notation (EN) [23][24].

RDFS [25] provides a data modelling vocabulary for concepts, such as class, subclass, domain, and range. It can be utilized for creating simple ontologies on top of RDF. OWL [26] extends RDFS with a more comprehensive vocabulary for modeling complex ontologies.

A semantic reasoner infers logical consequences from a set of explicitly asserted facts or axioms. Semantic inference discovers new relationships based on the data and additional information in the form of a vocabulary, e.g., ontology and a set of rules. HermiT [27], Owlgres [28], Pellet [29], and Jena Framework [30] are well-known semantic reasoners. Jena is a Java framework for building semantic applications with a rule-based inference engine to perform reasoning based on RDFS and OWL ontologies.

Semantic reasoning has been proposed for IoT systems. CoBrA [31][32] and Semantic Space [33] are two early approaches that utilize semantic technologies to enable context-awareness in small scale IoT systems. They both have context brokers built on top of Jena and context ontologies, which provide common vocabulary to model local IoT systems. More recent broker-centric approaches for semantic interoperability in IoT include Smart-M3 [34][35] and INSTANS [36][37]. They combine semantic technologies with publish-subscribe architectures to provide multi-device, multi-domain, and multi-vendor interoperability in IoT.

μJena is one of the first tools to manage ontologies and RDF stored in mobile devices [38]. LOnt implements Jena API for mobile devices [39]. Gu et al. proposed a mobile framework for ontology processing and reasoning. The reasoner contains a forward chaining rule-based inference engine, but it only supports a subset of OWL ontology inference rules [40]. Similarly, μOR [41] reasoner and "MiniOWL and MiniRule" [42] reasoner reason over a subset of OWL entailments. AndroJena [43] and Apache Jena on Android [44] provide Android based mobile devices with semantic reasoning capabilities.

As a first step of applying semantic reasoning in edge computing, Vazquez et al. propose the Smart Objects Awareness and Adaptation Model (SoaM) but this research focuses on a very limited set of reasoning capabilities. Ontological and semantic approaches for recognizing complex human activities from sensor data on edge devices have been proposed [45][46]. In our early research [4][47][23][49][50], we proposed solutions to enable semantic data encoding and information sharing with resource-constrained devices in the IoT context. In this paper, we go beyond the state of the art by extending semantic reasoning capabilities to edge nodes and analyzing the performance of data transferring and semantic reasoning in IoT systems with edge devices.

III. System Design

This section describes the requirements, scenarios, rules and data that guided designing of the edge architecture and the experiments.

A. Design Requirements

Aiming to fully support semantic reasoning for IoT systems with edge nodes, we emphasize four requirements in our experimental IoT system as follows.

Scalability. A big amount of heterogeneous devices are connected to IoT systems. To fulfill the scalability requirement, the IoT system should be able to process a big amount of dynamically generated data from IoT devices.

Heterogeneous data processing. As IoT devices work with different operating systems, employ different semantic annotation methods, and utilize different semantic formats, the IoT

system should cope with various communication mechanisms and different modules to process semantic data.

Balance of Computation. The computation workload of the whole IoT system should be balanced to guarantee the quality of service. For example, some applications require low latency services. The IoT system should allow enough computing and communication resources in some IoT nodes to cope with the heavy workload and latency requirements.

Semantic data processing and knowledge extraction. The IoT system should support popular RDF syntaxes such as RDF/XML, JSON-LD, and N3. Moreover, a mechanism is required to access the ontologies and rule sets from Cloud to enable mobile devices to preform various reasoning tasks [51].

B. Scenario and data

Our scenario is about a transportation system in a smart city. The taxi cabs around the city of Oulu are equipped with GPS and related software. Real taxi trajectories have been collected with this system. The taxis deliver the information to our experimental IoT system for decision making.

The raw data is in XML format and stored in SQLite database. When the GPS sensor of a taxi generates a new value, we store the data as an individual observation. In our experiments, we use the following eight properties of the observations: observation record ID, data timestamp, area ID, location (longitude and latitude), velocity, driving direction, and taxi ID.

The original data set contains 65,000 separate taxi trajectories formed by 5,543,348 observations (72,063,524 RDF statements). To study scalability, we generate from this data set a new data set of 200 taxis driving in the city during the same period. Figure 1 presents main concepts of the static OWL ontology for our smart transportation use case. Other relations, such as properties, are excluded in this figure. This lightweight static ontology is loaded on Cloud and edge nodes. IoT systems often reason from highly dynamic data generated from heterogeneous devices with static knowledge, because it is an efficient solution to deduce results and to keep reasoning sound and complete.

As presented in Table I, we implement 29 semantic rules to deduce 16 different activities of cars, inducing low and high speed, traffic jams, speeding, long stops, turning left and right and making U-turns, accelerating and decelerating strongly, and areas where taxis stop often for a while. More complex rules can be formed by combining these basic rules. The reasoner deduces facts from a sequence of observations by comparing consecutive values of direction, velocity, timestamp and location with forward chained rules. Rules are used in an incremental manner, which enables reasoning of all required knowledge from a sequence of observations. Incremental rules enable the distribution of reasoning tasks. For example, a right turn is assumed to happen after a taxi has driven at a relatively low speed, say, lower than 25 km/h, and if the direction change is near 90 degrees [47].

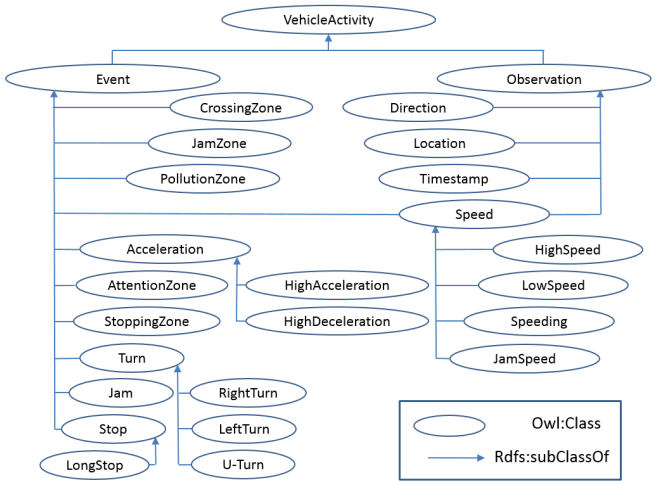

Fig. 1. High level static ontology for semantic reasoning in transportation system use case.

IV. Architectures

We design and evaluate two architectures. In both architectures, the smart systems of taxi cabs, i.e. IoT nodes, receive data from hardware and transform it into RDF. The nodes deliver the data, depending on the architecture, either to the Cloud or to the edge devices.

The first architecture, "Cloud Reasoning Architecture" (CRA), places a semantic reasoner on the Cloud (Figure 2). The IoT nodes encode the raw data into four alternative syntaxes of RDF model: RDF/XML, JSON-LD, N3, and short EN format [23]. Short EN format [23] compresses the data size by replacing constant information with templates and prefixes and we add one extra step to transform short EN to Turtle. IoT nodes send the RDF data to Cloud through TCP/IP protocol. A semantic reasoner, an ontology respository, a knowledge base and a MQTT [48] server are located in the Cloud. The semantic reasoner receives the RDF data and performs rule-based reasoning tasks. Reasoning results, which include the individual RDF data with new properties, are stored in the knowledge base. MQTT server realizes publish-subscribe communications. This architecture simply connects IoT devices to the Cloud, where the knowledge base and all semantic reasoning tasks are located. This is a typical architecture in most current IoT systems.

We introduce edge nodes in the second architecture, "Edge Reasoning Architecture" (ERA) (Figure 3). The edge nodes are devices physically near IoT nodes and they have reasoning capability. The edge devices support encoding and decoding of all four RDF syntaxes. Because of their resource limitations, we only deploy lightweight reasoning tasks in edge nodes. The selected rules and a lightweight ontology are designed and an Android Jena [43] reasoner is implemented in the edge nodes. The edge nodes communicate with the Cloud utilizing MQTT and with IoT nodes utilizing socket.

TABLE I
SEMANTIC RULE SET (SLIGHTLY MODIFIED FROM[47])

Fact	Triggering rule
Low speed	Observation hasVelocity<25km/h → ns:LowSpeed
Jam	LowSpeed hasDuration>90s ∧ LowSpeed hasAverageSpeed<20km/h → ns:Jam
Long stop	LowSpeed hasVelocity<3km/h → Stop ∧ Stop hasDuration>3min → ns:LongStop
High speed	Observation hasVelocity>80km/h → ns:HighSpeed
Speeding	HighSpeed hasVelocity>100km/h → ns:Speeding
Left turn	LowSpeed[1] hasDirection(a) ∧ LowSpeed[2] hasDirection(b) ∧ a=b-90deg ∨ a=b+270deg → ns:LeftTurn
Right turn	LowSpeed[1] hasDirection(a) ∧ LowSpeed[2] hasDirection(b) ∧ a=b+90deg ∨ b=a-270deg → ns:RightTurn
U-Turn	LowSpeed[1] hasDirection(a) ∧ LowSpeed[2] hasDirection(b) ∧ a=b-180deg ∨ b=a+180deg → ns:U-Turn
High acceleration	Observation[2] hasVelocity(v2) hasTmeStamp(t2) and (v2-v1)/(t2-t1)>2.5m/s^2 → ns:HighAcc
High deceleration	Observation[2] hasVelocity(v2) hasTmeStamp(t2) and (v1-v2)/(t2-t1)>2.5m/s^2 → ns:HighDeacc
Crossing Zone	LeftTurn hasLocation(x) ∧ RightTurn hasLocation(x) → ns:CrossingZone
Stopping Zone	LongStop[1] hasLocation(x) ∧ LongStop[2] hasLocation(x) ∧ LongStop[3] hasLocation(x) → ns:StoppingZone
Jam Zone	Jam[1] hasLocation(x) ∧ Jam[2] hasLocation(x) ∧ Jam[3] hasLocation(x) → ns:JamZone
Pollution Zone	HighAcc[1] hasLocation(x) ∧ HighAcc[2] hasLocation(x) ∧ HighAcc[3] hasLocation(x) → ns:PollutionZone
Attention Zone	HighDeacc[1] hasLocation(x) ∧ HighDeacc[2] hasLocation(x) ∧ HighDeacc[3] hasLocation(x) → ns:GoSlowZone
U-Turn Zone	U-Turn[1] hasLocation(x) ∧ U-Turn[2] hasLocation(x) ∧ U-Turn[3] hasLocation(x) → ns:U-TurnArea

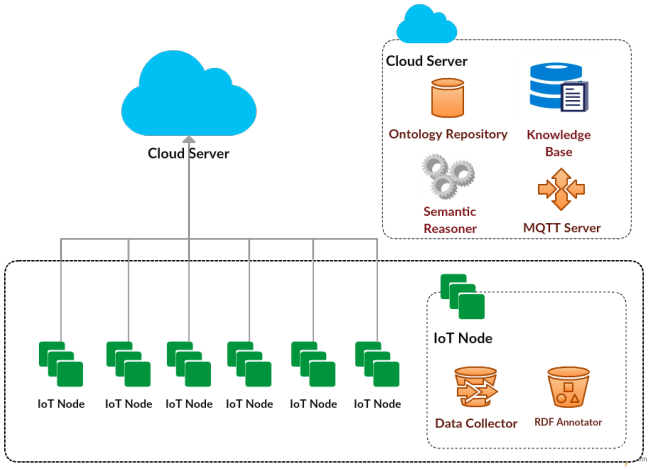

Fig. 2. An architecture for performing semantic reasoning on the Cloud.

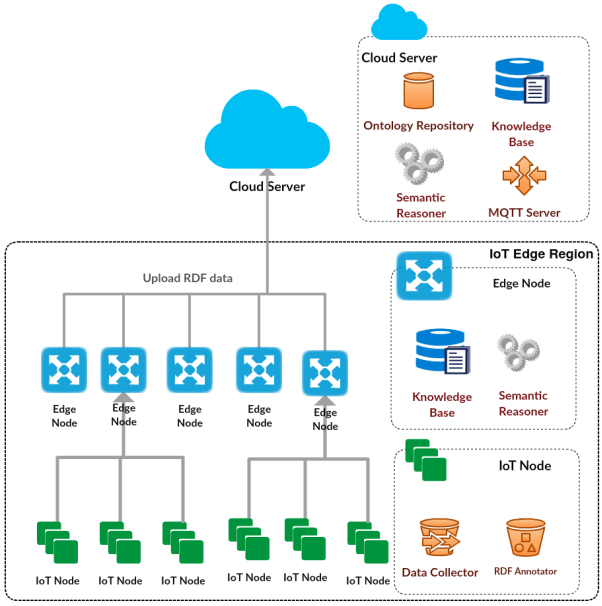

Fig. 3. An architecture for performing semantic reasoning on the edge nodes and Cloud.

V. EXPERIMENT AND ANALYSIS

A. Experiment Setup

To ease the system performance study, we separate reasoning into measurable independent steps.

1) IoT Node: IoT nodes have three simple functions: data collection, data encoding to RDF syntax, and data delivery. A PC computer replays the real data collected from taxi cabs (HP Desktop PC Elite Desk, Intel Core i5 4590 with 3.30 GHz CPU, 8GB memory). With this approach, we are able to evaluate two architectures with the same data set. IoT nodes simulate the software components of taxi cabs. 20-150 IoT nodes are executed simultaneously in the PC using threads. As IoT nodes generate data at a high frequency, they use cache to first store a certain amount of data (50 individual RDF statements) and then to deliver it as one message.

2) Edge Node: We utilize Android phones as edge nodes (LG Nexus 5X, Android OS 6.0.1, 4 Quad-core 1.44 GHz Cortex-A53 processors and 2 dual-core 1.82 GHz Cortex-A57 processors, Qualcomm MSM8992 Snapdragon 808 chip-set, 2GB RAM, 32GB storage). We utilize at maximum ten edge nodes; each deployed on an Android mobile phone.

The edge nodes receive data from the IoT nodes, perform local semantic reasoning tasks, and send the resulting data to the Cloud. In a basic mode, the edge nodes store an ontology locally, execute semantic reasoning, and send the data to the Cloud server simultaneously. In a second mode, the edge nodes fetch an ontology from a remote ontology repository server [51]. RDF data is stored with Android Jena framework and a rule-based hybrid rule engine is developed with Android Jena.

3) Cloud Server: The Cloud Server is deployed on Amazon EC2 Cloud platform (Amazon M4 Deca Extra Large Cloud, 160 GB memory, 124.5 EC2 compute units). One Amazon EC2 compute unit provides the equivalent CPU capability to

TABLE II
SEMANTIC REASONING EXPERIMENT TEST CASE FOR CRA

Group	No.	IoT nodes Number	RDF per node	Total RDF data
A	1	20	400	8000
	2	40	400	16000
	3	60	400	24000
	4	80	400	32000
	5	100	400	40000
B	6	40	800	32000
	7	60	533	32000
	8	80	400	32000
	9	100	320	32000
C	10	60	533	32000
	11	90	355	32000
	12	120	266	32000
	13	150	213	32000
D	14	60	200	12000
	15	60	400	24000
	16	60	600	36000
	17	60	800	48000

TABLE III
SEMANTIC REASONING EXPERIMENT TEST CASES FOR ERA

Group	No.	Edge node Number	nodes per Edge node	RDF per node	Total RDF Data
A	1	2	10	400	8000
	2	4	10	400	16000
	3	6	10	400	24000
	4	8	10	400	32000
	5	10	10	400	40000
B	6	4	10	800	32000
	7	6	10	533	32000
	8	8	10	400	32000
	9	10	10	320	32000
C	10	6	10	533	32000
	11	6	15	355	32000
	12	6	20	266	32000
	13	6	25	213	32000
D	14	6	10	200	12000
	15	6	10	400	24000
	16	6	10	600	36000
	17	6	10	800	48000

1.0-1.2 GHz 2007 Xeon processor. This 64-bit system has maximum bandwidth of 4000 Mbps. The server is physically located in Frankfurt, Germany. We implement Jena semantic reasoner and MQTT server for receiving data from edge nodes and IoT nodes in the Cloud.

4) Test cases: We design 17 semantic reasoning test cases on CRA (Table II) and ERA (Table III). For CRA, the variable "IoT node number" defines the total number of IoT nodes and "RDF per node" the amount of RDF statements which are delivered from one IoT node. For ERA, the variables "Edge node number" defines the number of the edge nodes and "nodes per edge node" the amount of IoT nodes which connect to an edge node. For example, in test case No.1 on ERA, we have two edge nodes and 20 IoT nodes; both edge nodes connect to ten IoT nodes. Each IoT node sends 400 observation RDF statements and there are 8000 observation RDF statements in total.

B. Experiments

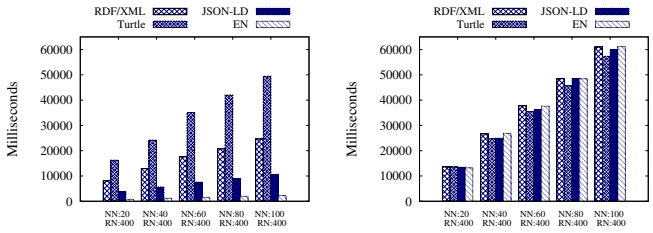

Fig. 4. Scalability results for group A (left:transferring, right:reasoning).

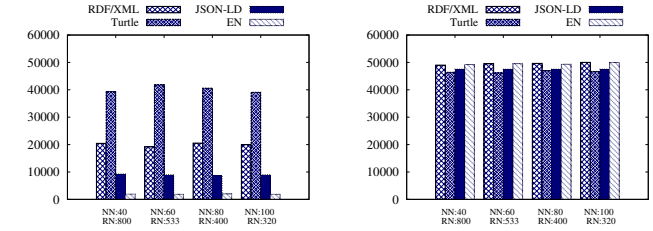

Fig. 5. Scalability results for group B (left:transferring, right:reasoning).

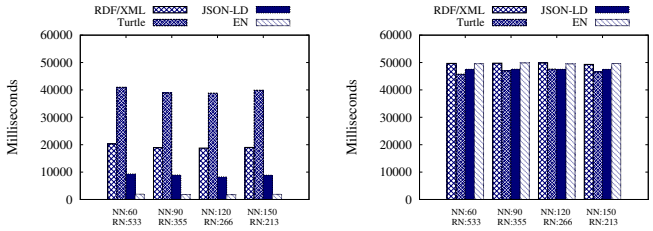

Fig. 6. Scalability results for group C (left:transferring, right:reasoning).

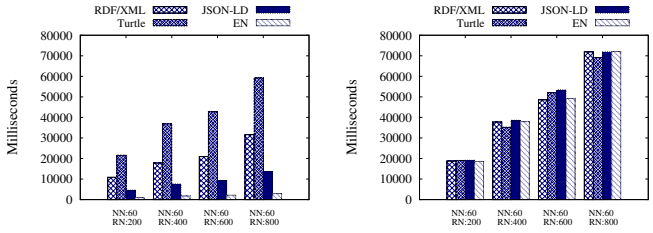

Fig. 7. Scalability results for group D (left:transferring, right:reasoning).

1) Scalability: Four RDF formats are compared in CRA: RDF/XML, Turtle, JSON-LD, and short EN. Data transferring latency from IoT nodes to Cloud server is measured. Reasoning latency is calculated at the Cloud server. As IoT nodes generate real time data, we ignore the data generating time and only focus on the total data transferring and reasoning time. The measured transferring time starts from building the MQTT client to set up the connection and ends with receiving the response. We use a response message to confirm that the information is delivered successfully to the Cloud. At the Cloud server, we measure the time of performing semantic reasoning tasks and storing data to the RDF database storage. The measurement starts from building a Jena model and ends with finishing the storage of the inferred facts in RDF.

The left figures in Figure 4, Figure 5, Figure 6, and Figure

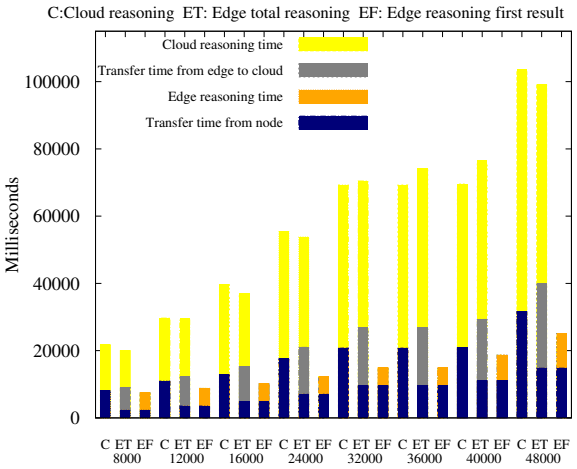

Fig. 8. Reasoning performance comparison between two architectures with RDF/XML.

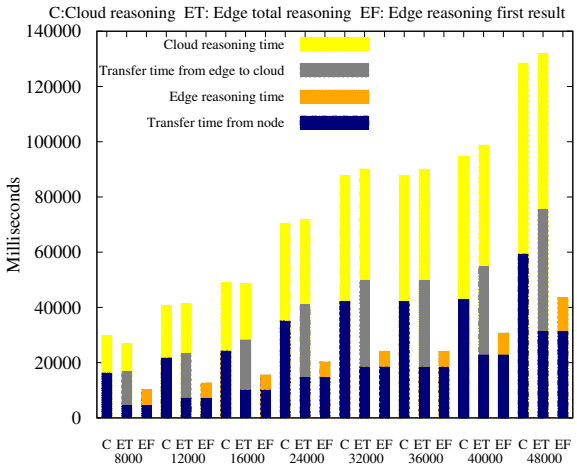

Fig. 9. Reasoning performance comparison between two architectures with JSON-LD.

7 present the data transferring times of group A, B, C, and D. The figures on the right present the reasoning times. The X-axis represents the number of IoT nodes and the number of RDF statements per node. Total data size is calculated from the number of IoT nodes and the number of RDF statements per node. For example, "NN:20 RN:400" means 20 IoT nodes collect data and each IoT node sends 400 individual observation as RDF statements. In this case, the total number of RDF statements is 8000. The Y-axis represents time in milliseconds. The data transferring time increases with the total RDF data size. From the data formats, short EN format consumes the shortest time, The JSON-LD is the second shortest, and the Turtle format is the longest one. For example, in the test case of 24000 RDF statements (No.3 in Group A), the transferring times of JSON-LD, RDF/XML, and Turtle are on the average 4.7, 10.7, and 21.2 times of that of EN, respectively. Examining all 17 sets of data in all four groups produces very similar results. The transferring times of JSON-LD, RDF/XML, and Turtle are on the average 4.6, 10.4, and 21.0 times of that of EN, respectively. Jena reasoning includes building Jena models and performing reasoning tasks. Regarding to reasoning time comparison, data with different formats shows a comparable performance for the same amount of data. The total reasoning time presents linear growth when the data size increases.

2) Comparison of CRA and ERA: In the second experiment, we compare CRA and ERA by measuring the latency of the complete data delivery and reasoning processes. We measure data transferring times from the IoT nodes, reasoning times at Cloud server, reasoning times at the edge nodes, and the data transferring time from the edge nodes to Cloud. The edge nodes only perform semantic reasoning with two selected rules, i.e. "High Acceleration" and "High De-acceleration". Other semantic rules are performed only on the Cloud. We utilize the same four RDF formats.

Figure 8, Figure 9, Figure 10, and Figure 11 present the results. We present results in each test case in three columns:

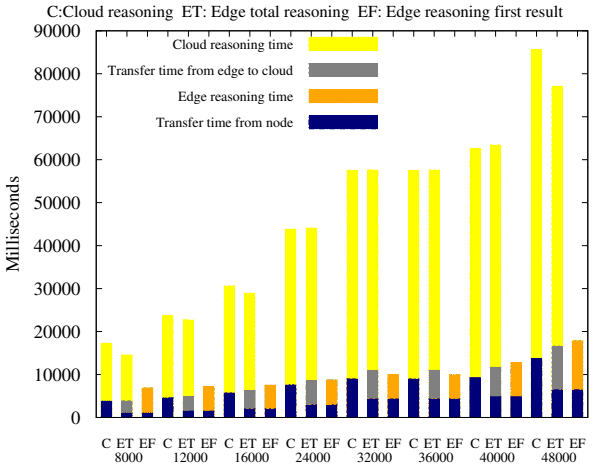

Fig. 10. Reasoning performance comparison between two architectures with Turtle.

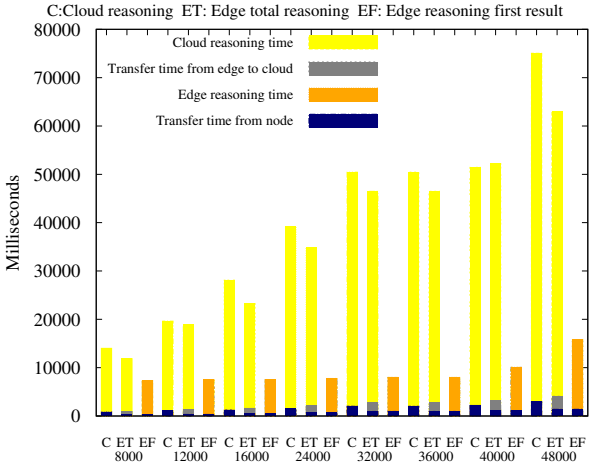

Fig. 11. Reasoning performance comparison between two architectures with short EN.

the left column (C) represents the total reasoning time of the CRA; the middle column (ET) represents the total reasoning time of the ERA; and the right column (EF) represents reasoning time of the ERA for generating first results. The edge nodes perform a part of the reasoning tasks and we call the result from edge nodes as "first result". The latency for generating first results include the time to transfer data from IoT nodes to edge nodes and the reasoning time of performing semantic reasoning tasks on edge nodes. We choose eight out of seventeen experiments based on the total RDF data size. The X-axis of the figures represents the size of the RDF data. Our experiments show that the first results are ready much earlier than the results from Cloud server. For example, in the third test case with 16000 RDF statements in short EN experiments, the Cloud reasoning time is ten times of that of the first results. The average ratio between Cloud reasoning time and first results time from edge nodes for RDF/XML is 5.1 times, for Turtle is 4 times, for JSON-LD is 6.6 times, and for EN is 8.9 times. When utilizing computation capabilities of edge nodes, the average Cloud reasoning time reduces 12.4% for RDF/XML format, 12.3% for Turtle format, 6.2% for JSON-LD format, and 12.1% for EN format. The total transferring time on edge of Turtle is 1.2 times of reasoning time in average and the time of EN is 6% of the average reasoning time.

3) Edge reasoning performance comparison with different rule sets: Aiming to study task distribution strategies between the Cloud and edge nodes that lead to the best overall performance, the last experiment focuses on the performance improvement of ERA with deploying different semantic reasoning rules on edge nodes. It's obvious that the more rules are processed on edge nodes, the fewer rules are processed on the Cloud. We choose RDF/XML as the only data format. In this experiment, there are eight edge nodes and each edge node will collect data from ten IoT nodes. Each IoT node will send 400 individual RDF data to either edge node or Cloud server. Thus for CRA, 80 IoT nodes will send 32000 RDF data in total.

We design four groups of task distributions: Group A implements all reasoning tasks on the Cloud server; Group B implements rules which related to "High Average Speed" on the edge nodes and the rest on Cloud; Group C implements rules related to "High Acceleration" and "High De-acceleration" on edge nodes and the rest on Cloud; Group D implements rules related to both "High Average Speed", "High Acceleration" and "High De-acceleration" on the edge nodes and the rest on the Cloud.

Figure 12 presents the result of the performance comparison between different task distributions on edge nodes. In each group, the overall time consumption is:

$$T_{Overall} = \max\left(T_{Cloud}, T_{Edge}\right)$$

In Figure 12, the left chart shows the overall processing time including both data transferring time and reasoning time and the right chart shows only the reasoning time. The Cloud column represents the total processing time from IoT node to edge nodes and from edge nodes to Cloud server. The edge

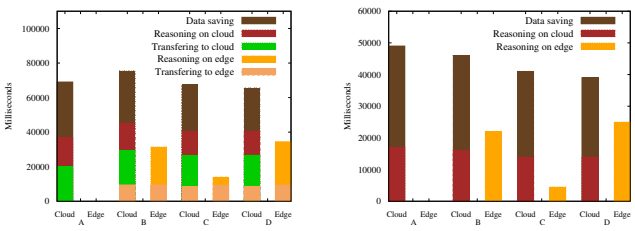

Fig. 12. Performance comparison with different rule set. (left: overall data processing time, right: reasoning time)

column only counts the processing from IoT nodes to edge nodes including data receiving time and reasoning time. For example, in Group B, IoT nodes first send the data to edge nodes, and then the edge nodes simultaneously reason the data and send data to the Cloud. Thus both columns have the same orange part, which is data sending time from IoT nodes to edge nodes. Then the yellow part in edge column represents the reasoning time on edge. The green part represents the data sending time from edge to Cloud. And the red part represents the data reasoning time on Cloud and the brown part represents the data storage time on Cloud. As the Cloud and edge are simultaneously performing tasks, we could calculate the overall reasoning time with the formula:

$$T_{Reasoning} = \max\left(T_{CloudReasoning}, T_{EdgeReasoning}\right)$$

From the right chart we observe that the more rules implemented on the edge node, the less reasoning time needed from Cloud server. Comparing with Cloud reasoning in Group A, Group B reduces 4% reasoning time, Group C reduces 15.6% reasoning time and Group D reduces 20.7% reasoning time. From the left chart, comparing with Cloud reasoning time in Group A, Group B increases 8% reasoning time, Group C reduces 1.4% reasoning time and Group D reduces 5% reasoning time.

C. Analysis

We conclude that the transferring time is relevant to syntaxes. The required transferring time scales linearly with the payload size, which depends the data structures and formats. In our experiments, we send different amounts of similar data structures. Therefore, the encoding ratios between different formats are similar with different amounts of data. Regarding reasoning latency, different RDF syntaxes require significantly different amount of time in building Jena models but require the same amount of time for reasoning after they are loaded in a model. Hence, the more time is required in building Jena models, the more time is needed for reasoning. In general, the amount of reasoning time grows linearly as RDF statements are added.

The transferring time is based on the total data size and network status. We measured the total size of the 50 individual data and observed that:

$$Size_{Turtle} > Size_{RDF/XML} > Size_{JSON-LD} > Size_{ShortEN}$$

The comparison of CRA and ERA shows that adding edge nodes in IoT systems accelerates data processing and reduces need for network bandwidth. When only first results are required, the ERA can generate results ten times faster than the CRA. As our IoT edge devices are located in Finland and the Cloud Server is in Germany, the long distance and unstable network affect the latency. The networking equipments affect the general performance as well. We are using panOULU [52], which has five types of wireless routers with different capabilities. In our experiments, we try to utilize the maximum capacity of the network.

From the third experiment, distributing semantic reasoning tasks, we summarize two strategies for optimizing the design of the edge based IoT systems. First, to achieve fast responses, transferring time to the edge and reasoning time on the edge devices should be minimized. Second, to achieve minimum overall reasoning time, reasoning on edge and processing time on Cloud (including transferring time to Cloud, reasoning time on Cloud and storing to database) need to be balanced. In other words, selecting the correct amount of semantic rules and deploying a suitable amount of reasoning tasks improve the overall performance. Distributing workload on the edge shortens processing time in the Cloud.

How much time is saved depends on the system structure and capabilities of edge nodes. The reasoning time can be decreased when the edge devices have sufficient processing capability and the reasoning task can be distributed into several edge nodes and executed in a parallel manner. However, even when the reasoning time is decreased, the overall time can be increased. As we observe from Figure 8, Figure 9, Figure 10, and Figure 11, overall reasoning time for generating complete results are comparable. Some of the experiments even show that ERA requires more overall reasoning time than CRA, for example in 32000, 36000, and 40000 RDF statements bars of Figure 8. This is because adding an edge device adds also one transfer operation; instead of sending data from an IoT node to the Cloud, the data is first sent to the edge nodes and then from the edge nodes to the Cloud. Overall time is saved only when the reduced reasoning time is larger than the increased transferring time. When IoT devices have poor network connections, for example moving vehicles, data transfer requires more time and thus undertaking more tasks on edge nodes improves performance.

VI. Discussion

Edge and fog computing allow computation to be performed at the edge of the network, on downstream data on behalf of Cloud services and upstream data on behalf of IoT services [3]. This article focuses on analyzing the performance of semantic reasoning in IoT systems with edge nodes. Our contribution is a detailed analysis of three experiments with a large smart transportation data set to address the research challenges of scalability and latency. For studying the influence of edge computing, we evaluate the performance of semantic reasoning with cloud and edge architectures. Moreover, we evaluate the performance with different amount of tasks deployed on the edge nodes.

The size of the RDF data is an essential factor for data transferring and storage. Thus, selecting a proper format for RDF data can improve the performance of semantic processing. However, our results show that the same format may perform differently on different cases. The time to transfer semantic data is closely related to the RDF data sizes and formats, but the semantic reasoning times for different syntaxes are similar.

Our study shows that adding edge nodes into an IoT system can improve system performance: first results can be generated faster, bandwidth usage on the core network can be reduced and the workload of the Cloud can be reduced as well. Physical proximity between edge nodes and IoT nodes improves transferring efficiency. Additional edge nodes reduce the reasoning time of the Cloud server. If computation is distributed properly, the overall processing time is reduced, as reasoning tasks are executed at the edge in a parallel manner. The degree of improvement depends on the relationship between the transferring, reasoning, and storing times.

This research focuses on analyzing the performance of an edge based IoT system. We deploy the system on the Amazon EC2 Cloud platform and Android devices. Our performance evaluation is based on measuring latency for data transfer and reasoning. However, the results may differ in other hardware and software systems. Our experiment does not count how many background services are running during our test. More objective measurement metrics could be selected in the future research, such as the number of executed CPU instructions for specific processes. Similarly, the measurement of transferring time of the RDF data is also affected by network situation. The comparison between CRA and ERA is based on Oulu taxi scenario and we utilize predefined rules and ontologies. More experiments could be carried out to study whether the result differ when the data, rules and ontologies are more dynamic. Moreover, challenges such as how to assign the tasks on edge nodes and what criteria should be adopted to optimize the performance should be addressed in future research [54]. Finally, we will evaluate the resource usage of semantic reasoning and investigate minimum required resources for semantic reasoning on edge nodes.

Acknowledgment

We thank Professor Claudio Bettini and Dr. Ilias Gerostathopoulos for their advice for this paper. The first author would like to thank Jorma Ollila Grant for funding this research.

References

[1] A. Sheth, C. Henson, and S. Sahoo, "Semantic Sensor Web," IEEE Internet Computing, July/August 2008, pp. 78–83.

[2] CISCO, "Fog computing and the internet of things: Extend the cloud to where the things are", https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf [2015].

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges", IEEE Internet of Things Journal, Vol. 3, Iss. 5, pp. 637–646, 2016.

[4] X. Su, P. Li, Y. Li, H. Flores, J. Riekki, and C. Prehofer, "Towards semantic reasoning on the edge of IoT systems," in Proc. the 6th International Conference on the Internet of Things, 2016, pp. 171–172.

[5] IDC FutureScape, "Worldwide Internet of Things 2016 Predictions," https://www.idc.com/getdoc.jsp?containerId=259856 [Nov. 2015].

[6] RDF 1.1 Primer, https://www.w3.org/TR/rdf11-primer/.

[7] E. Ahmed and M.H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," Future Generation Computer Systems, 2016.

[8] NOKIA, "Radio applications cloud servers," https://www-03.ibm.com/press/us/en/pressrelease/40490.wss [2014].

[9] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," IEEE Internet of Things Journal, Vol. 3, Iss. 6, pp. 854–864, 2016.

[10] M. Vögler, J. Schleicher, C. Inzinger, S. Nastic, S. Sehic, and S. Dustdar, "LEONORE - Large-scale provisioning of resource-constrained IoT deployments," in Proc. the 9th IEEE International Symposium on Service-Oriented System Engineering, IEEE SOSE, 2015, pp. 78–87.

[11] M. Vögler, J. M. Schleicher, C. Inzinger, and S. Dustdar, "A Scalable Framework for Provisioning Large-Scale IoT Deployments," ACM Transactions on Internet Technology, Vol. 16, Iss. 2, pp.1–20, 2016.

[12] N.K. Giang, M. Blackstock, R. Lea, and V.C.M. Leung, "Developing IoT applications in the Fog: A Distributed Dataflow approach," in Proc. the 5th International Conference on the Internet of Things, 2015, pp. 155–162.

[13] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Replisom: Disciplined Tiny Memory Replication for Massive IoT Devices in LTE Edge Cloud," IEEE Internet of Things Journal, Vol. 3, Iss. 3, pp. 327–338, 2016.

[14] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the internet of things: early progress and back to the future," International Journal on Semantic Web and Information Systems (IJSWIS), Vol. 8, pp. 1–21, 2012.

[15] X. Su, J. Riekki, J.K. Nurminen, J. Nieminen, and M. Koskimies, "Adding Semantics to Internet of Things," Concurrency and Computation: Practice and Experience, Vol. 27, Iss. 8, pp 1844–1860, 2015.

[16] RDF/XML Syntax Specification, https://www.w3.org/TR/REC-rdf-syntax/ [2014].

[17] JSON for Linking Data, https://json-ld.org/.

[18] RDF 1.1 N-Triples, https://www.w3.org/TR/n-triples/.

[19] R. Cyganiak, A. Harth, and A. Hogan, "N-quads: Extending n-triples with context," 2008.

[20] RDF 1.1 Turtle, https://www.w3.org/TR/turtle/.

[21] RDFa 1.1 Primer, https://www.w3.org/TR/xhtml-rdfa-primer/.

[22] Notation3 (N3): A readable RDF syntax, https://www.w3.org/TeamSubmission/n3/.

[23] X. Su, "Lightweight Data and Knowledge Exchange for Pervasive Environments," Ph.D. Thesis, University of Oulu, Faculty of Information Technology and Electrical Engineering, Acta Universitatis Ouluensis series C581, 2016.

[24] X. Su, J. Riekki, and J. Haverinen, "Entity Notation - Enabling Knowledge Representations for Resource-Constrained Sensors," Personal and Ubiquitous Computing, Springer, Vol. 16, Iss. 7, pp. 819–834, 2012.

[25] RDF Schema 1.1, https://www.w3.org/TR/rdf-schema/.

[26] P. Hitzler, M. Krötzsch, B. Parsia, P.F. Patel-Schneider, and S. Rudolph, "Owl 2 web ontology language primer," https://www.w3.org/TR/owl2-primer/.

[27] R. Shearer, B. Motik, and I. Horrocks, "Hermit: A highly-efficient owl reasoner," International Workshop on OWL: Experiences and Directions, Vol. 432, 2008, pp. 91–101.

[28] M. Stocker and M. Smith M, "Owlgres: A scalable owl reasoner," International Workshop on OWL: Experiences and Directions, vol. 432, 2008.

[29] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," Web Semantics: science, services and agents on the World Wide Web, Vol. 5, Iss. 2, pp. 51–53, 2007.

[30] B. McBride, "Jena: A semantic web toolkit," IEEE Internet computing, Vol.6, Iss. 6, pp. 55–59, 2002.

[31] H. Chen, T. Finin, A. Joshi, L. Kagal, F. Perich, and D. Chakraborty, "Intelligent agents meet the semantic Web in smart spaces," Internet Computing IEEE, Vol. 8, Iss. 6, pp. 69–79, 2004.

[32] H. Chen, T. Finin, and A. Joshi A, "Semantic Web in the Context Broker Architecture," in Proc. 2nd IEEE Annual Conference on Pervasive Computing and Communications (PerCom), 2004, pp. 277–286.

[33] X. Wang, J. Dong, C. Chin, S. Hettiarachchi, and D. Zhang, "Semantic Space: An Infrastructure for Smart Spaces," IEEE Pervasive Computing, Vol. 3, Iss. 3, pp. 32–39, 2004.

[34] J. Honkola, H. Laine, R. Brown, and O. Tyrkko O, "Smart-M3 Information Sharing Platform," in Proc. IEEE Symposium on Computers and Communications (ISCC), Riccione, Italy, 2010, pp. 1041–1046.

[35] J. Kiljander and A. D'elia, "Semantic interoperability architecture for pervasive computing and internet of things," IEEE Access, 2, pp. 856–873, 2014.

[36] H. Abdullah, M. Rinne, S. Törmä, and E. Nuutila, "Efficient Matching of SPARQL Subscriptions using Rete," in Proc. the 27th Annual ACM Symposium on Applied Computing, New York, NY, USA, ACM, 2012, pp. 372–377.

[37] M. Rinne, E. Nuutila, and S. Törmä, "INSTANS: High-Performance Event Processing with Standard RDF and SPARQL," in Proc. International Semantic Web Conference (ISWC), Posters and Demonstrations Track, Boston, USA, 2012.

[38] F. Crivellaro, "μJena: Gestione di ontologie sui dispositivi mobili," MSC Thesis of Politcnico di Milano.

[39] M. Koziuk, J. Domaszewicz, R.O. Schoeneich, M. Jablonowski, and P. Boetzel, "Mobile context-addressable messaging with dl-lite domain model," D. Roggen, C. Lombriser, G. Tröster, G. Kortuem and P. Havinga (ed) Smart sensing and context, Zürich, Switzerland, Springer Berlin Heidelberg, pp. 168–181, 2008.

[40] T. Gu, Z. Kwok, K.K. Koh, and H.K. Pung, "A mobile framework supporting ontology processing and reasoning," in Proc. the 2nd workshop on requirements and solutions for pervasive software infrastructures, in conjunction with the 9th international conference on ubiquitous computing. Innsbruck, Austria, 2007, pp.16–19.

[41] S. Ali and S. Kiefer, "μOR - Micro a micro owl dl reasoner for ambient intelligent devices," Abdennadher N and Petcu D (eds) Advances in grid and pervasive computing, Geneva, Switzerland, Springer Berlin Heidelberg, pp. 305–316, 2009.

[42] J.I. Vazquez, "A reactive behavioural model for context-aware semantic devices," Doctoral Dissertation of Universidad de Deusto, 2007.

[43] Androjena, Porting of Jena to Android, https://github.com/lencinhaus/androjena. Cited 2016/06/29.

[44] Apache Jena on Android, https://elite.polito.it/research/downloads/182-jena-on-android-download.

[45] J. Ye, G. Stevenson, and S. Dobson, "USMART: An Unsupervised Semantic Mining Activity Recognition Technique," ACM Trans. Interact. Intell. Syst., Vol. 4, No. 4, Art. 16, 2014.

[46] D. Riboni, T. Sztyler, G. Civitarese, and H. Stuckenschmidt, "Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning," In Proc. of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, New York, USA, 2016, pp. 1–12.

[47] A.I. Maarala, X. Su, and J. Riekki, "Semantic Reasoning for Context-aware Internet of Things Applications," IEEE Internet of Things Journal, Vol. 2, Iss. 4, pp. 1–13, 2016.

[48] Message Queuing Telemetry Transport (MQTT), http://mqtt.org/.

[49] J. Kiljander, A. Ylisaukko-Oja, J. Takalo-Mattila, M. Eteläperä, and J. Soininen, "Enabling semantic technology empowered smart spaces," Journal of Computer Networks and Communications, 2012.

[50] J. Kiljander, F. Morandi, and J. Soininen, "Knowledge sharing protocol for smart spaces," International Journal of Advanced Computer Science and Applications, Vol. 3, Iss. 9, pp.100–110, 2012.

[51] X. Su, P. Li, H. Flores, J. Riekki, X. Liu, Y. Li, and C. Prehofer, "Transferring Remote Ontologies to the Edge of Internet of Things Systems," The 12th International Conference on Green, Pervasive and Cloud Computing, Cetara, Italy, May 2017, pp. 538–552.

[52] T. Ojala, J. Orajärvi, K. Puhakka, I. Heikkinen, and J. Heikka J, "panoulu: Triple helix driven municipal wireless network providing open and free internet access," in Proc. the 5th International Conference on Communities and Technologies, ACM, 2011, pp. 118–127.

[53] C. Negus and T. Boronczyk, CentOS Bible, Wiley Publishing, 2009.

[54] H. Flores, P Hui, P Nurmi, E Lagerspetz, S Tarkoma, J Manner, V Kostakos, Y Li, and X Su, "Evidence-aware Mobile Computational Offloading", IEEE Transactions on Mobile Computing, 2017.