ConTaaS: An Approach to Internet-Scale Contextualisation for Developing Efficient Internet of Things Applications

Ali Yavari
RMIT University, Australia
mail@aliyavari.com

Prem Prakash Jayaraman, Dimitrios Georgakopoulos
Swinburne University of Technology, Australia
pjayaraman, dgeorgakopoulos@swin.edu.au

Surya Nepal
CSIRO, Australia
surya.nepal@data61.csiro.au

Abstract

The Internet of Things (IoT) is a new internet evolution that involves connecting billions of sensors and other devices to the Internet. Such IoT devices or IoT things can communicate directly. They also allow Internet users and applications to access and distil their data, control their functions, and harness the information and functionality provided by multiple IoT devices to offer novel smart services. IoT devices collectively generate massive amounts of data with an incredible velocity. Processing IoT device data and distilling high-value information from them presents an Internet-scale computational challenge. Contextualisation of IoT data can help improve the value of information extracted from IoT. However, existing contextualisation techniques can only handle small datasets from a modest number of IoT devices. In this paper, we propose a general-purpose architecture and related techniques for the contextualisation of IoT data. In particular, we introduce a Contextualisation-as-a-Service (ConTaaS) architecture that incorporates scalability improving techniques, as well as a proof-of-concept implementation of all these that utilises elastic cloud-based infrastructure to achieve near real-time contextualisation of IoT data. Experimental evaluations validating the efficiency of ConTaaS are also provided in this paper.

1. Introduction

The Internet of Things (IoT) supports the development of smart services via the discovery and integration of Internet-connected IoT devices (e.g. sensors, smart phones, etc.) and their data. It is estimated that by 2025 [1, 2] the number of IoT devices will reach 50 billion. With such a tremendous increase in number of IoT devices getting connected to the Internet, IoT presents a novel Big Data challenge to the Internet. For example, the emergence of IoT has created opportunities for development of IoT services for smart factories, smart farms, and smart cities. Such IoT applications often require the ability to understand the context of data collected from Internet-based sensors and use that to support timely and effective decision making. For example, consider an IoT application that utilises IoT data collected from a specific point of interest (e.g. a specific suburb of Melbourne) rather than an entire city or country. Contextualisation of IoT data permits filtering out data collected from other points of interest, and hence reduces the amount to data that needs to be processed further. A more general description of context can be articulated on the basis of a widely used definition of context by Dey et al. [17]: *Context* or *contextual information is any information about any entity that can be used to effectively reduce the amount of reasoning required (via filtering, aggregation, and inference) for decision making within the scope of a specific application*. Contextualisation is then the process of identifying the data relevant to an entity based on the entity's contextual information.

Contextualisation excludes irrelevant data from consideration and has the potential to reduce data from several aspects including volume, velocity, and variety in IoT applications and subsequently improve the data processing and knowledge extraction in IoT applications.

The ability to contextualise the stored or streamed Internet-scale data from millions and billions of devices is a grand challenge. Traditional data management approaches such as relational databases lack the efficiency required for applications that deal with Big Data [4, 5]. Similarly, recent high-performance processing techniques for Big Data, such as MapReduce [6], are not ideal for IoT applications because they fall short in supporting IoT's real-time and incremental data processing requirements [7, 8]. Moreover, to the best of our knowledge there is no specified architecture for contextualisation of Big Data reported in the literature.

Related work includes a plethora of techniques for adaptation of data based on the interest(s) of an

HICSS

application and the use of context information in context-aware computing, ubiquitous computing and so forth [9, 10]. However, virtually all these techniques are application dependent. Furthermore, existing contextualisation techniques are not scalable and will not support Internet-scale IoT applications.

In this paper, we propose a novel architecture and related techniques for the contextualisation of Internet-scale IoT data (we collectively refer to all these as Contextualisation-as-a-Service or ConTaaS). The ConTaaS architecture is realised over scalable and distributed cloud computing services that provide scalable contextual data processing as needed by IoT applications. A proof of concept implementation of ConTaaS is also described later in the paper.

The main contributions of this paper include the following:

- A novel contextualisation architecture, which we refer to as *ConTaaS Architecture*, for contextualising Internet-scale IoT data and facilitating the developing of efficient IoT applications.
- A novel contextualisation technique, which we refer to as *ConTaaS contextualisation*, that employs prime factorisation to scale up the contextualisation of IoT data.
- A cloud-based ConTaaS implementation that utilises commercially available cloud infrastructure services (more specifically Amazon EC2).

The remainder of this paper is organized as following: Section 2 reviews the background and the state-of-the-art in contextualisation. Section 3 describes the proposed contextualisation-as-a-Service (ConTaaS). Section 4, presents an experimental scenario for ConTaaS Architecture. Section 5, describes the design of a sample ConTaaS implementation and presents an evaluation of ConTaaS. Finally, the conclusion and future work are presented in Section 6.

2. Background

In this section, we briefly discuss related background to IoT contextualisation from the following three perspectives:

- Sensors and data semantics
- Context aspects
- Querying sensor data

Over the past 10-15 years, sensors have been used in several different areas, such as environment monitoring, traffic control, healthcare, etc. Sensors are typically small devices which are capable of sensing, storing, and transmitting data, as well as actuating over wired and wireless networks. One of the main challenges in sensor networks is to transform the data coming from heterogeneous sensing devices that are manufactured by different vendors and for different applications into homogeneous, discoverable, and usable information presented in human and machine readable format. There have been several recent efforts to tackle this challenge by meta-data tagging or semantic annotating sensor data [11]–[13]. While meta-data is any sort of informal information attached to the data, semantically annotated data is associated with ontologies [14] that expressively and formally define and describe the type, properties and interrelationships of the data. Semantically annotated data not only is more understandable, but it can also be reasoned to deduce new knowledge and subsequently increase the expressiveness of the data. The Sensor Web Enablement (SWE) [15] standard from Open Geospatial Consortium is an international effort to standardize all types of sensors, transducers and sensor data repositories accessible and discoverable via the Internet. SWE consists of the following: 1) The Sensor Model Language that includes a standard model and an XML Schema for describing sensor characteristics, specification and capabilities, such as the location of the sensors. 2) The Observation and Measurements standard model and schema for describing observations and measurements from sensors and sensor networks. 3) The Observation interface for entering queries and retrieving observation and sensory data.

SWE standards and XML schemas are able to describe sensor data and observations with meta-data in some extent but they do not support semantic reasoning, abstraction and classification provided by semantic technologies. The Semantic Sensor Network (SSN) [16] adds semantics describing sensors and sensor networks. The SSN ontology is compatible with SWE and extends semantic support for SWE. SSN ontology expressively represents sensor and observations of the environment.

From the context perspective, several researchers attempted to define and use context in developing intelligent applications in areas ranging from ubiquitous and mobile computing, to artificial intelligence [17]–[20]. Context has been introduced by several researchers in the literature. The most common definition in the literature is by Dey et al. [17] who define context as "*any information that can be used to characterise the situation of an entity. An entity is a*

person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." While this definition is sufficient for context-aware computing it does not necessarily capture context from the perspective of IoT or other large-scale and multi-application environments. Moreover, there is no clear separation between data and context in this definition. Context in IoT is closely aligned with the notion of context in context-aware computing due to the fact that context-aware computing and IoT have similarities in terms of data. However, they do not have the same scale and resource processing demands.

A partially overlapping notion of context has been defined for query processing in databases. To explain this we first note that the structure of storing data in a database is a database model and query is a syntax based on a formal language to access the data. Several alternative data models have been proposed in the literature including flat file, hierarchical, star schema, graph databases and so forth. The most common and well-established database model is the relational database model and SQL [21]. Relational databases are based on first-order predicate logic and are sets of predicates over a finite set of predicate variables. However, relational databases require sophisticated resource to deal with complex queries and data structures. Moreover, relational databases are not necessarily designed to communicate with other external data sources and are limited to adapt their schema frequently to the new structures. Non-relational databases (also referred as noSQL or nonSQL) provide data models that are more suitable for Big Data, distributed and scalable[1] data storage [22]. The Resource Description Framework (RDF) [23] provides a data model and structure to represent data and is the most standardized noSQL database model. RDF represents data with three parts including Subject, Predicate and Object that are commonly referred as Triple. Figure 1, depicts a sample Triple.

RDF generally can be represented as RDF/XML, N-Triples, Turtle, Notation3 and N-Quads. SPARQL (SPARQL Protocol and RDF Query Language) [24] is a standardized query language for RDF triples. Figure 2, depicts a SPARQL example. SPARQL scalability has been discussed previously [25] and it is believed that due to its performance and scalability, it can be

used for Internet-scale applications. However, SPARQL queries are complex in comparison with SQL and relational database.
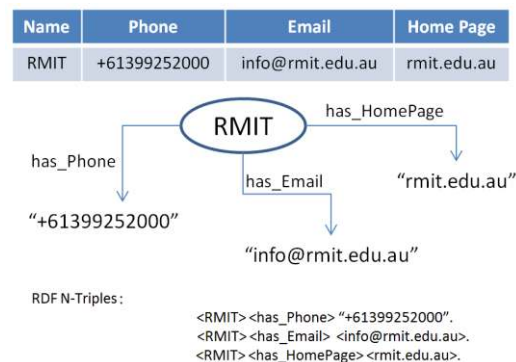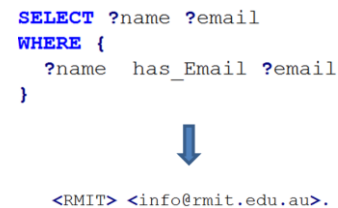

Figure 1. Triples

Figure 2. SPARQL

3. Contextualisation of IoT Data in ConTaaS

We define contextualization as the processes of contextually filtering, aggregating, and inferring (contextual operations) data using context. Contextualisation of the internet- scale data is hard as it requires algorithms that can process large volumes of heterogeneous data arriving at very high velocity. In addition, contextualisation involves mapping and scheduling contextualisation tasks on cloud resources and other high performance data processing infrastructure. Due to increasing number of devices in IoT, scalability is an important challenge. Therefore, contextualisation of IoT data should be scalable in such a way that the IoT data input volume, complexity and variety can be handled by the available computing resources. Several solutions and techniques are discussed in the literature that can perform contextualisation to some extent, but they are mostly designed and developed for a particular application and can neither be generalized nor scaled up. In the following sections, we conceptually define the process of contextualisation of Internet-scale IoT data. First we

---

[1] There are typically two broad categories for scaling database systems. Horizontal scaling is by adding more nodes such as adding a new server or data center. Vertical scaling is adding resource to the current server such as increasing processor or memory. Relational databases are compatible with vertical scaling while there are limited in horizontal scaling

will present some of our definitions that are used to represent context using semantic representation.

3.1. ConTaaS concepts and definitions

Definition 1 (Triples): A triple is a statement describing a piece of data in form of three parts including <*Subject*, *Predicate*, *Object*>. *Subject* is the identifier of the entity that the data is describing. *Object* is the description of the *Subject* in terms of the relation described in *Predicate*. For example, a triple <RMIT University, hasEmail, info@rmit.edu.au>, describes that RMIT University (*Subject*) has an email address (*Predicate*) which is info@rmit.edu.au (*Object*).

Definition 2 (RDF Triples): A RDF triple is a *formal* triple in such a way that the *Subject* can be a blank-node or Internationalized Resource Identifier (IRI) [26], the *Predicates* are only IRI and *Object* can be IRI, literals or blank-node. A blank-node in RDF is a node in such a way that it does not contain any data, but it groups data as a parent node [Figure 3].
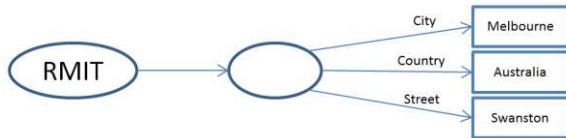

Figure 3. Blank node

Definition 3 (Context): In this paper we represent context as triples where Subject is a specific *application*, Predicates describe the *relevancy* of the entity with the information and the object is the *information*. For example, the context (triple)

<*App1, RestaurantType, vegetarian*>

represents that App1 is interested in vegetarian restaurants. Subsequently, we briefly define context as any combinations of the predicates and objects that are relevant to a given application.

Definition 4 (Contextualised Data): Contextualised data for a particular application is a subset of RDF triples that are filtered, aggregated and inferred according to the context relevant to the given application.

Definition 5 (Context ID): Context ID (CID) is a label assigned to each particular context(s) to uniquely represent the context. ConTaaS assigns an exclusive prime number to each context triple (i.e., to each combination of a particular predicate and a particular object).

Definition 6 (Application Context ID): Application Context ID (ACI) is a label that represents all the contextual preference of a given application. Contextual preference is a set of contexts that are relevant to a given application defined by the application or the user of the application. ACI is not unique for each particular application and can dynamically change based on the applications and their changing contextual preferences. In this paper, we compute and assign the ACI number to each application by multiplying the Context IDs relevant to this particular application. For example, assume that App1 is an application that has two contexts as:

<*App1, Location, Melbourne*>, CID=7
<*App1, RestaurantType, Vegetarian*>, CID=29

ACI number of these two contexts will be 203, i.e., the factor of 7 and 29. Therefore, ACI numbers identify:

1) The contexts of a given application
2) Applications with similar contextual preferences

For any application A with an ACI number n we will have:

$$n = \prod_{i=1}^{w(n)} p_i$$

In the above equation, $w(n)$ identifies the number of contexts relevant to application A, and each distinct prime factors p_i of n is one of the CIDs relevant to that *Subject*. For example, in order to derive contexts of an application with ACI = 77, by prime factorization of 77 we will have 7 and 11 that are relevant CIDs, and the number of contexts ($w(77)$) is 2. If ACI number is a prime number it indicates that *Subject* has only one context.

Definition 7 (Contextual Query): Contextual Query is a query that considers CIDs or ACIs in the query text. In the rest of this paper N-triples queries are referred to as queries and N-quads queries are referred to as Contextual Queries.

3.2. ConTaaS Operations

Earlier in this paper we defined contextualisation by referring to three primary contextual operations, namely: *filter*, *aggregate*, and *infer*. The filter operation applies to an input and the output is a subset of the input that satisfies the condition. This condition

for contextualisation is whether the data has any contextual relevancy with any application. Filter operation does not modify the data and it only indicates if the data should be considered in queries from the application or not. The aggregation operation receives several inputs and mathematically or statistically processes them to compute the output. For example, in a room with multiple temperature sensors an aggregation operation can calculate the average as a representation of the room temperature. Aggregation in contextualisation is any mathematical operation that can merge two or more input triples into a single output triple. The infer operation is more complex and it is used to deduce new knowledge from the input data. For example, if RMIT is in Melbourne and Melbourne is in Australia, we can deduce that RMIT is in Australia.

In the following paragraphs we describe these contextual filtering, aggregation, and inference operations in more detail:

Contextual Filter: Processing all the data collected by all the sensors is not possible due to the limitations and scalability issues in computing resources. The Contextual Filtering operation labels triples in such a way that only triples that are relevant to at least one application will be labelled. Contextual Filter converts N-triples to N-quads by adding another part to triples which is the CID. In this way, any N-triples can be excluded from the Contextual queries because they are not contextually relevant to any particular application. Contextual Filter can use any labelling approaches as long as labels are uniquely defined. In this paper we use prime numbers as previously described.
The CID calculated for triples indicates:

- Triples that satisfy contextual preferences of a given application.
- Triples that identically satisfy the same contextual preferences

Contextual Aggregation: This operation aggregates two or more context triples based on similarity in context preferences of the applications. For example, consider the following triples:

<App3, Location, Melbourne, 7>
< App3, Symptom, "Headache", 3>
<App3, Symptom, "Pain", 5>
< App2, Symptom, "Headache", 3>
<App2, Symptom, "Pain", 5>
<App1, Food, "Vegetarian", 2>

In this set of triples there is no particular application interested in both "Headache" or "Pain" individually.

However, if there are applications that are interested in both of them at the same time we generate other triples with blank nodes [Figure 3] with new CIDs as follows:

<App3, Location, Melbourne, 7>
< App2, Aggregated, _:b, 11>
< App3, Aggregated, _:b, 11>
< _:b, Symptom, "Pain", 3>
< _:b, Symptom, "Headache", 5>
<App1, Food, "Vegetarian", 2>

Then, we update CIDs of the triples that satisfy all the aggregated contexts with the appropriate blank node.

Contextual Inference: Inference is the process of deducing new knowledge. Contextual inference takes contexts as input to deduce new knowledge. For example, suppose that application App1 has ACI number 210 and application App2 has ACI number 30. We can infer that all the contextual data relevant to App2 are also relevant to App1. Furthermore, dividing 210 by 30 and prime factorization of the result we can infer that the data relevant to App2 are relevant to App1 if they also satisfy CID=7.

3.3. ConTaaS Architecture

Figure 4, presents the proposed ConTaaS Architecture. Contextualisation using the depicted ConTaaS Architecture is performed in a sequence of steps. In the first step, the raw data from IoT devices are annotated using semantic representations such as the Semantic Sensor Network Ontology (SSN) [15]. This aspect is independent of the underlying semantic framework used and is capable of supporting any other metadata representation schemes (e.g. SensorML). Semantically annotated IoT data is then converted to RDF triples and then stored for further processing. Further processing involves: 1) The *application context* that is specified and represented in the system as *domain context* (this may also include user context such as user preferences), and 2) the filter, aggregate, and infer operations we described in section 3.2. The output of this step is the contextualised data that is presented to the application.

4. Experimental Scenario

In this paper, we describe our proposed architecture in a Smart city scenario. Smart cities rely on current advances in technologies, such as IoT, networking, data analytics, recommendations, and decision support, to deliver better quality of life to citizens. The smart city vision is made up of many building blocks around
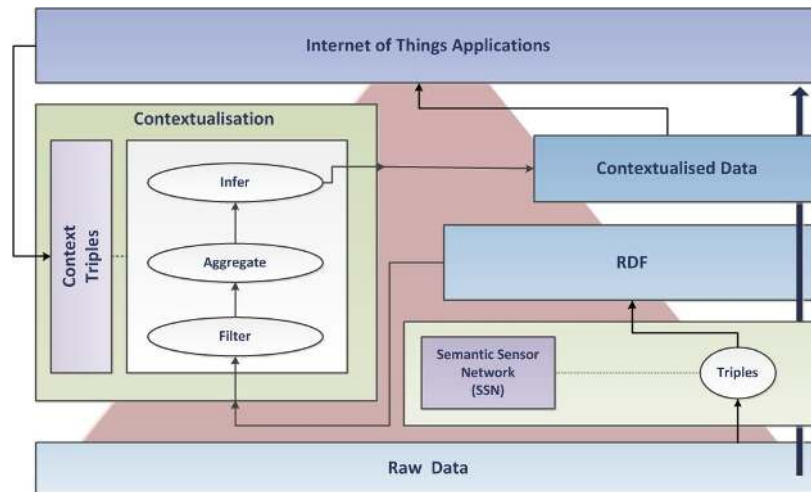
Figure 4: ConTaaS Architecture

a combination of application and services such as smart health, smart grid, smart traffic control, smart transport system, etc. Although, the ConTaaS architecture proposed in this paper is general-purpose and can be used to realise IoT contextualisation in any domain, in this section we focus on a smart health application. Consider an outbreak of Ebola virus disease that originated in March 2014 from the west of Africa. After 5 months and more than 4500 death reports, the World Health Organization declared this outbreak an international public health emergency. To stop Ebola virus transmission that occurs via physical contact, it is necessary to do the following: 1) diagnose the virus as soon as possible, 2) isolate the patients by limiting contact with other people, and finally 3) start infection control and treatment. Ebola virus disease's most common symptoms are fever, fatigue, loss of appetite, vomiting, diarrhoea and headache [27]. Speeding up diagnosis by identifying any person who has all or most of these symptoms and determining if this person has been travelling in a high-risk area during a particular time period could be potentially lifesaving. In particular, countries dealing with Ebola must solve the following problems:

1. Check all the residents or travellers to their countries to see if they have been in Africa during the first few months of 2014 and if they have the symptoms.
2. Determine if those who were identified in (1) had any physical contact with anybody known to be infected.
3. Transfer those identified in (2) to a hospital.

Just like Ebola, the World Health Organization has also identified Zika as an international public health emergency. Zika mainly transfers with a bite of infected Aedes mosquito and have symptoms such as fever, conjunctivitis, joint pain and skin rash. Zika was originally considered to be a mild disease [28]. However, recent scientific research showed that Zika virus can cause microcephaly on unborn babies of mothers who are infected by the virus during their pregnancy. The first step in Zika mitigation is similar to (1) for Ebola, as we need to know if any residents or tourists of any country has visited Brazil or other Zika high-risk areas and at the same time they manifested the Zika symptoms. However, as Zika is a mild virus and does not have any particular treatment, the only concern for infected people is to rest and avoid pregnancy until the virus disappears completely from the body, which is approximately a 6 month period.

With current advances in mobile smart phone and wearable technology, we assume the possibility of collection data from people including their location [29] (with due concerns to privacy and security). Additionally, assume that we have the records of the symptoms of people that are manually entered into a database via health applications, collected from hospitals and during medical checks, or from sensors and wearable devices such as smart watches. Such data from citizens will be massive. Furthermore, for any particular application we may need to frequently repeat the data analysis process on the entire dataset. Managing a dataset of this kind is resource demanding and its analysis requires sophisticated computing resources.

ConTaaS has the potential to solve this problem by reducing the complexity of the data analysis query and extract valuable knowledge from such data. In this scenario we describe two data analysis applications for Ebola and Zika namely EbolaApp and ZikaApp. Subsequently we define the context triples for
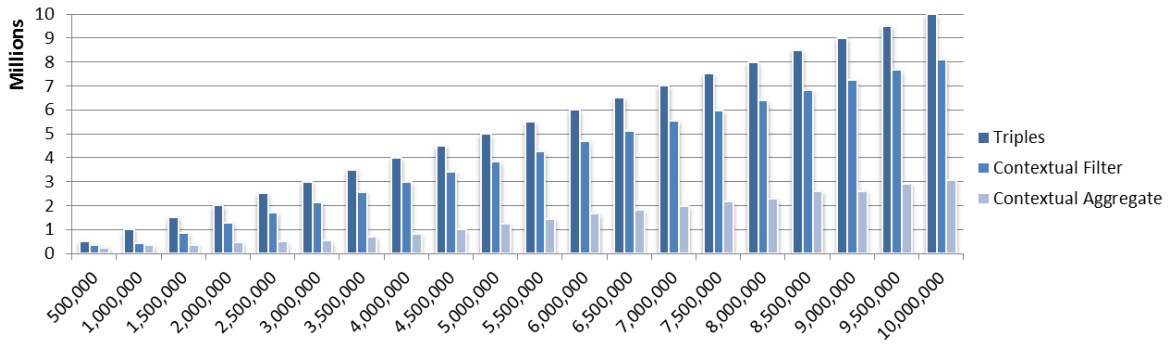
Figure 8. Data Reduction

EbolaApp as follows:

- <EbolaApp, Location, Australia >
- <EbolaApp, Symptom, "fever">
- <EbolaApp, Symptom, "fatigue">
- <EbolaApp, Symptom, "loss of appetite">
- <EbolaApp, Symptom, "vomiting">
- <EbolaApp, Symptom, "diarrhoea" >
- <EbolaApp, Symptom, "headache">
- <EbolaApp, Visited, Africa>

ZikaApp context triples include:

- <ZikaApp, Location, Australia>
- <ZikaApp, Symptom, "fever">
- <ZikaApp, Symptom, "conjunctivitis">
- <ZikaApp, Symptom, "joint pain" >
- <ZikaApp, Symptom, "skin rash">
- <ZikaApp, Visited, Brazil>

Table 1, shows the data records from a sample of 5 persons. The CIDs of data records [Figure 5] have been assigned by the Contextual Filter. Ava does not have any CIDs. Lack of CIDs indicates that this person is not relevant to any of the two applications. The next step, which is performed via Contextual Aggregation and based on Aggregated CIDs [Figure 6], determines that Sophia will not need to be considered further as she is not relevant to EbolaApp or ZikaApp.
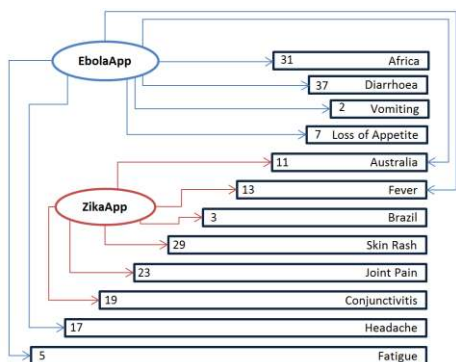
Finally, the ACI number for these applications are calculated as EbolaApp = 5863 and ZikaApp = 6721. The ACI numbers for remaining persons are computed (i.e. John=143, Jacob=6721 and Emily=611), and based on this, we determine that Jacob is the only person that satisfies the ZikaApp context and no one satisfies the EbolaApp context.
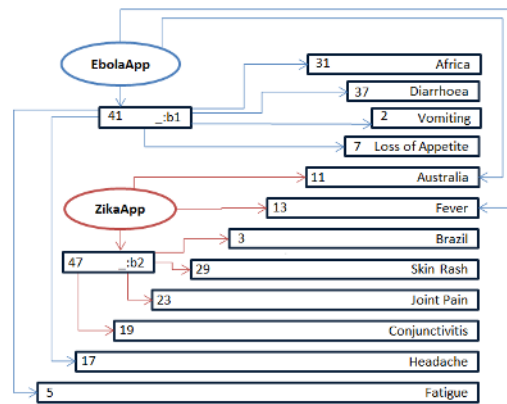

Figure 6. Aggregated CIDs

Contextual Inference can determine that Emily is a suspect for a Zika infection as the only contexts she does not meet is the fact that her current location is not Australia. Subsequently, a list of triples with ACI=611 can be used in Australia's borders to detect suspicious passengers.
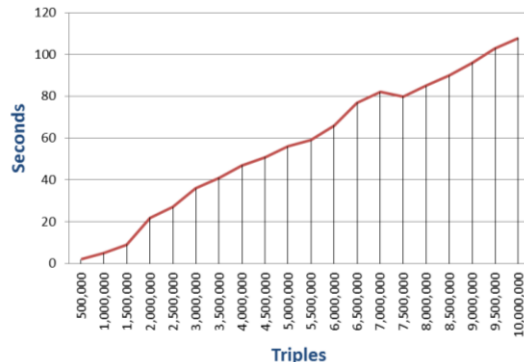

Figure 5. CIDs

Figure 7. Contextualisation Process Time

Table 1. Sample Experimental data

Name	Symptom	Visited	Location	CID	Aggregate CIDs
John	fever, fatigue, loss of appetite , diarrhoea	Africa, Sweden	Australia	13,7,5,37,31,11	13, 11
Sophia	diarrhoea, vomiting, headache	France	Germany	37, 2, 17	
Ava	sore throat, cough	Germany	Canada		
Jacob	conjunctivitis, skin rash, fever, joint pain, headache	Brazil, Italy	Australia	19, 3, 29, 13, 23, 17, 11	47,13,11
Emily	conjunctivitis , joint pain, skin rash, fever	Brazil, Australia	Japan	19, 23, 29, 3, 13	47,13

5. Implementation and Evaluation

We developed a proof-of-concept implementation of ConTaaS and evaluated its performance on an "m4.xlarge" instance of Amazon's Elastic Cloud Computing (EC2) platform [30] running the Ubuntu 14.04 operating system. The hardware configuration of the servers included four vCPUs 3.3GHZ Intel Xeon processors and 16 GB of RAM memory. To implement and run the ConTaaS test-bed we employed Java version 1.8, Apache Jena [31] and PHP 5.2. Our dataset included synthetic triples generated randomly based on the previously described experimental scenarios. Our evaluation assumed that the data points have been generated by IoT sensors deployed on patients in hospitals and/or data collected by medical personnel.

Figure 7, shows the performance of contextual filter when processing different numbers of triples. Figure 8, shows the number (data reduction) of triples after each operation.

6. Conclusion and Future Work

Scalable and real-time contextualisation of IoT data has the potential to significantly improve the value of information extracted and data processing efficiency for large scale IoT applications. In this paper, we proposed a general-purpose architecture and techniques for scalable contextualisation of IoT data. We also performed experimental evaluation of the proposed architecture and related techniques using synthetic data from a smart health application to validate their ability to collect and process high-volume IoT sensor data.

Though, the volume of Internet-scale data would be larger than the dataset used in this evaluation, the performance outcomes indicate that the proposed ConTaaS architecture and contextualisation techniques will scale well for Internet-scale datasets. In future work, we aim to investigate dynamic contexts and the application of MapReduce to distributed contextualisation operations. The architecture described in this paper is designed to be incremental, but in this paper we didn't address incremental functionalities of the algorithms. We also plan to investigate more complex reasoning by using other ontologies.

7. References

[1] G. Says, "6.4 Billion Connected' Things' Will Be in Use in 2016, Up 30 Percent From 2015," *Gart. Inc*, 2015.

[2] C. V. N. I. Cisco, "Global mobile data traffic forecast update, 2013–2018," *White Pap.*, 2014.

[3] A. K. Dey, "Understanding and using context," *Pers. Ubiquitous Comput.*, vol. 5, no. 1, pp. 4–7, 2001.

[4] S. Kaisler, F. Armour, J. A. Espinosa, and W. Money, "Big data: Issues and challenges moving forward," in *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, 2013, pp. 995–1004.

[5] S. Sagiroglu and D. Sinanc, "Big data: A review," in *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, 2013, pp. 42–47.

[6] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[7] V. Kalavri and V. Vlassov, "MapReduce: Limitations, optimizations and open issues," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, 2013, pp. 1031–1038.

[8] Y. Qin, Q. Z. Sheng, N. J. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: A data-centric view of the internet of things," *ArXiv Prepr. ArXiv14072704*, 2014.

[9] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *Commun. Surv. Tutor. IEEE*, vol. 16, no. 1, pp. 414–454, 2014.

[10] A. Whitmore, A. Agarwal, and L. Da Xu, "The Internet of Things—A survey of topics and trends," *Inf. Syst. Front.*, vol. 17, no. 2, pp. 261–274, 2015.

[11] V. Nebot and R. Berlanga, "Building data warehouses with semantic web data," *Decis. Support Syst.*, vol. 52, no. 4, pp. 853–868, 2012.

[12] A. Zimmermann, N. Lopes, A. Polleres, and U. Straccia, "A general framework for representing, reasoning and querying with annotated semantic web data," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 11, pp. 72–95, 2012.

[13] M. Serrano, H. N. M. Quoc, D. Le Phuoc, M. Hauswirth, J. Soldatos, N. Kefalakis, P. P. Jayaraman, and A. Zaslavsky, "Defining the stack for service delivery models and interoperability in the Internet of Things: a practical case with OpenIoT-VDK," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 4, pp. 676–689, 2015.

[14] S. Staab and R. Studer, *Handbook on ontologies*. Springer Science & Business Media, 2013.

[15] M. Botts, G. Percivall, C. Reed, and J. Davidson, "OGC® sensor web enablement: Overview and high level architecture," in *GeoSensor networks*, Springer, 2006, pp. 175–190.

[16] M. Compton, P. Barnaghi, L. Bermudez, R. GarcíA-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, and others, "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 17, pp. 25–32, 2012.

[17] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Handheld and ubiquitous computing*, 1999, pp. 304–307.

[18] A. Schmidt, M. Beigl, and H.-W. Gellersen, "There is more to context than location," *Comput. Graph.*, vol. 23, no. 6, pp. 893–901, 1999.

[19] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Ca4iot: Context awareness for internet of things," in *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, 2012, pp. 775–782.

[20] D. Baker, D. Georgakopoulos, H. Schuster, and A. Cichocki, "Awareness provisioning in collaboration management," *Int. J. Coop. Inf. Syst.*, vol. 11, no. 01n02, pp. 145–173, 2002.

[21] C. J. Date and H. Darwen, *A guide to the SQL Standard: a user's guide to the standard relational language SQL*, vol. 55822. Addison-Wesley Longman, 1993.

[22] J. Han, E. Haihong, G. Le, and J. Du, "Survey on NoSQL database," in *Pervasive computing and applications (ICPCA), 2011 6th international conference on*, 2011, pp. 363–366.

[23] G. Klyne and J. J. Carroll, "Resource description framework (RDF): Concepts and abstract syntax," 2006.

[24] E. Prud'Hommeaux, A. Seaborne, and others, "SPARQL query language for RDF," *W3C Recomm.*, vol. 15, 2008.

[25] M. Morsey, J. Lehmann, S. Auer, and A.-C. Ngonga Ngomo, "DBpedia SPARQL benchmark–performance assessment with real queries on real data," *Semantic Web–ISWC 2011*, pp. 454–469, 2011.

[26] M. Dürst and M. Suignard, "Internationalized resource identifiers (IRIs)," 2004.

[27] N. J. Beeching, M. Fenech, and C. F. Houlihan, "Ebola virus disease," *BMJ*, vol. 349, p. g7348, 2014.

[28] D. Gatherer and A. Kohl, "Zika virus: a previously slow pandemic spreads rapidly through the Americas," *J. Gen. Virol.*, vol. 97, no. 2, pp. 269–273, 2016.

[29] P. P. Jayaraman, C. Perera, D. Georgakopoulos, and A. Zaslavsky, "Efficient opportunistic sensing using mobile collaborative platform mosden," in *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on*, 2013, pp. 77–86.

[30] E. Amazon, "Amazon elastic compute cloud (Amazon EC2)," *Amaz. Elastic Compute Cloud Amaz. EC2*, 2010.

[31] A. Jena, "Apache jena," *Jena Apache Org Online Available Httpjena Apache Org Accessed Mar 20 2014*, 2013.