Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 52 (2015) 500 – 506

Procedia Computer Science

The 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015)

Big Data Storage in the Cloud for Smart Environment Monitoring

M. Fazio*, A. Celesti, A. Puliafito, M. Villari

University of Messina, C.da Di Dio - Sant'Agata, Messina 98166, Italy

Abstract

Monitoring activities detect changes in the environment and can be used for several purpose. To develop new advanced services for smart environments, data gathered during the monitoring need to be stored, processed and correlated to different pieces of information that characterize or influence the environment itself. In this paper we propose a Cloud storage solution able to store huge amount of heterogeneous data, and provide them in a uniform way. To this aim, we adopt an hyrid architecture that couple Document and Object oriented strategies, in order to optimize data storage, querying and retrieval. In this paper, we present the architecture design and discuss some implementation details in the development of the architecture within a specific use case.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Big Data; Storage system; Smart environment; Sensing; IoT; Cloud; SWE

1. Introduction

The growing exploitation of smart environments and audio/video streams is causing a massive generation of complex and pervasive digital data. Sensing equipment and sensor networks are deployed to monitor phenomena of interest providing many heterogeneous measurements and multimedia data. Then, data are stored, shared and processed for several purposes, such as healthcare[1], air quality monitoring[2], and risk management[3]. For many years, enterprise organizations have accumulated growing stores of data, running analytics on that data to gain value from large information sets, and developing applications to mange data exclusively. However, a new trend is arising, where data production, information management and application development are decoupled, thus giving to business companies different roles in the market. In such a scenario, flexible solutions to merge activities of vendors, manufacturers, service providers, and retailers are necessary. In this paper we focus the attention on data storage services, and we present a new storage architecture specifically aimed to monitoring activities in smart environment.

In the Internet of Things (IoT) perspective, billions of physical sensors and devices are interconnected through the Internet to provide many heterogeneous, complex and unstructured data. Many effort in the industry and in the research community have been focused on the storage of IoT data, in order to balance costs and performance for data maintenance and analysis[4]. Indeed, the design of powerful storage systems can efficiently handle the requirements

* Maria Fazio. Tel.: +39-090-3977344 ; fax: +39-090-3977176.
E-mail address: mfazio@unime.it

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2015.05.023

of big data applications and Cloud computing is expected to play a significant role in IoT paradigm. Indeed, Cloud storage offers huge amount of storage and processing capabilities in a scalable way[5]. Thus, we designed a monitoring-oriented Cloud architecture for the storage of big data, that can be exploited for the development of application and services useful in many different applications for smart environments (e.g., smart cities, homeland security, disaster prevention, etc.).

This paper analyzes Big Data issues arising from monitoring activities, and discusses different sotrage technologies that can be exploited to support different types of data, in order to optimize data storage, querying and retrieval. Our storage architecture couples both the Document and Object oriente Storage Systems approaches in Big Data storage, thus to provide a unique solution able to treat different information sources. Moreover, it exploits the Cloud computing technology to benefit of scalability and reliability. From the point of view of the Cloud user, data gathered from the monitoring infrastructure are provide in a uniform way, that has been designed according to the Sensor Web Enablement (SWE) specifications defined by the Open Geospatial Consortium (OGC)[6] .

The paper is organized as follows. Section 2 describes related works. Data features in smart environments are discussed in Section 3. In Section 4, we present our Cloud storage solution, discussing many design choices. A few implementation highlights are discussed in Section 5. Our conclusion are summarized in Section 6.

2. Related Works

New Cloud infrastructures interacting with Sensors and Internet of Things (IoTs) are recently appearing in literature. A Cloud Platform useful for supporting the *Fully Connected Car* system is presented by Dingo et al. [7]. The architecture is at very high level, in which telco and Cloud operators are included in the picture. A much more detailed Platform as a Service architecture is called *CloudThings*[8]. It represents a collection of Cloud services offered by the IT market (i.e., Facebook, GAE,...), smart devices and embedded systems(i.e., Wiring, Sun SPOT, mbed, Arduino) and Cloud applications (Heroku, Paraimpu,...). The implementation shows all adopted solutions tailored for Smart Home scenarios, a real use case deployed in Oulu Finnish city. Cloud4Sensing[3] is a framework that integrates two different strategies for managing sensing resources in the Cloud and let the end-user free to choose which type of Cloud service he needs. Specifically, the framework provides services according to a data-centric or a device-centric model: the former is implemented as a PaaS (Platform as a Service) able to abstract and store heterogeneous sensing/actuation data that are provided to clients; the latter is implemented as a IaaS (Infrastructure as a Service) offers a sensing/ actuation infrastructure to the clients. Another high level platform[9] is able to integrate Wireless Sensor Networks with Cloud Computing. All these platforms present the same type of functionalities and elements. In our view, for making real progress it is necessary to take into account interoperability among heterogeneous systems.

Cloud Computing is also becoming the basis for Big Data needs. At the Infastructure as a Service (IaaS) level, Big Data can leverage the Storage capabilities of Clouds, as well at the same time, it can rely on computation inside VMs[10]. Also Hadoop, installed into VMs, is optimized for processing Big Data. It is interesting to see that VM instances and their configurations strongly affect this kind of processing. Using Cloud resources in relation to Big Data task is a straightforward goal. Hadoop is the larger used opensource framework adopted for managing Big Data with Map/Reduce approach. Another example of Big Data processing in the Cloud is presented by Rao et al.[11]. In this work the computation framework used is Sailfish, a new Map/Reduce environment similar to Hadoop. Sailfish was conceive for improving the disk performance for large scale Map-Reduce computations. It tries to build network-wide data aggregation inside data centers and improve disk throughput. Big Data is driving the way of using algorithms and resources even in the Cloud. Big Data problem in e-health scenarios looks at NoSQL DBs as the key solution toward the full development of IoT, and specifically they investigate on how to shift towards the Web of Things[12].

The work described in our paper is based on SWE, the standard of OGC that is currently looking to form the "Sensor Web for IoT" Standards Working Group[6], able to explore opportunities to extend the SWE framework and to harmonize it with existing open standards to accommodate Web-friendly and efficient implementations of sensor interfaces and sensor networks using the REST protocol[13]. The problem to find an abstraction on sensing data representation was also identified from Ballarini et al.[14], where they analyzed the concepts of proximity, adjacency or containment. They even introduced the contexts of data representation with different dynamics. They provided a global model with a dynamic interoperability disregarding how the global view should be accomplished. Their

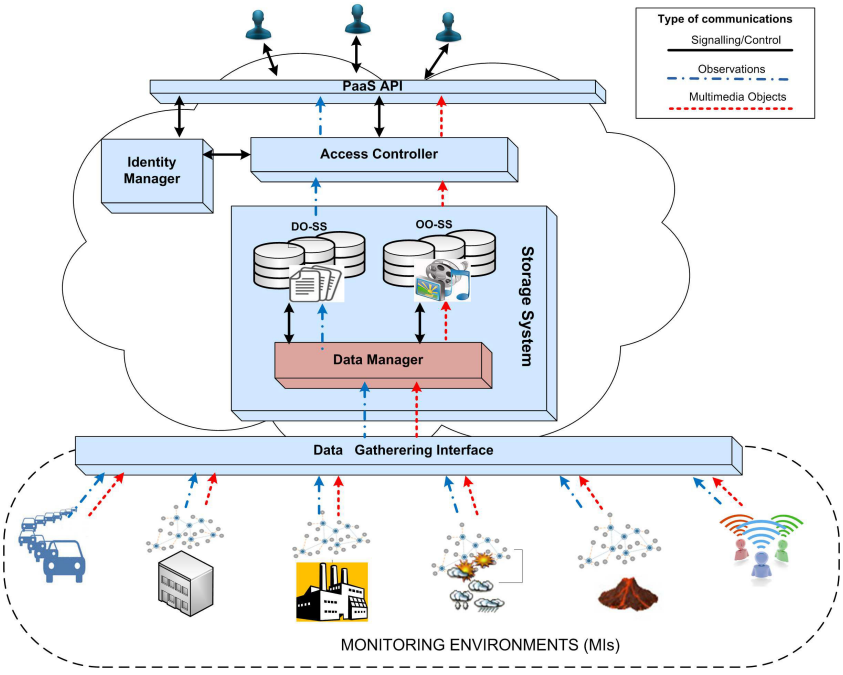

Fig. 1. The Cloud storage service

decision-maker is requested to process a huge amount of incoming data, but it is not clear how such a problem is practically addressed (i.e. scalability problems).

3. Big Data Issue in Smart Environments

The Cloud storage solution we present provides data access and query capabilities to several heterogeneous data sources. It allows users to express their needs in terms of type of measurement, time interval, geolocalization of data, etc., and to receive data according to a uniform format. Before presenting our solution, we need to present the main issues that need to be addressed in monitoring data management, thus to better explain our main design strategies.

Monitoring infrastructures in smart environments belong to different tenants spread on a worldwide area. There are several possible models that lead tenants to share their data over the Cloud. For example, the tenants provides data as open sensing data through the web. In this case, the Cloud storage provider is interested in integrating such type of data in its system; or the tenant is at the same time both resource provider and consumer, and it exploits the Cloud to extend his physical infrastructure by means of the Cloud virtual infrastructure; otherwise, the Cloud storage provider and the tenant company make commercial agreements.

The type of agreement between tenants of monitoring infrastructures and Cloud storage providers is out of the scope of this paper, but we want to highlight that, in a such a complex scenario, data coming from monitoring infrastructures are very heterogeneous. We can roughly classify such data in two main types:
1) *Observations*: measurements of physical or composed phenomena performed by sensing devices. Observations can be expressed by tuples (*key*, *value*) and stored in text file forwarded across the network;
2) *Objects*: multimedia contents (e.g., audio, image, video and animation) recorded by information content processing devices [15].

The meaning of ”Big Data” today deals with very large unstructured data sets (PetaByte of data [16]), that need of rapid analytics with answers provided in seconds. However, strategies to manage Big Data strongly depend on the specific type of data. Observations can generate Big Data because monitoring activities in a wide geographical area produce several tuples in short time interval. Thus, in long periods (days, months, years) a huge amount of data need

to be structured and stored. Observations can be made available through *Documents*, where they are encapsulated in a standardized internal format. An effective Document-Oriented Storage System (DO-SS) (e.g., MongoDB, Cassandra, CouchDB,...) indexes the contents of each document in order to make an easily retrieval of them. Moreover, a great deal of publishing is done in HTML, XML, JSON, or systems that can at least export or convert to those.

Objects can originate files with big size, but Big Data issues arise not only from the volume of Objects, but also with respect to their heterogeneous nature. Indeed, different types of queries can be executed to find an Object in a storage system according to the specific type of data. An Object-Oriented Storage System (OO-SS) (e.g., AWS S3, SWIFT, Kinetic,...) combines storage capabilities (e.g.,transparently persistent data, concurrency control, data recovery, associative queries,...) with object-oriented programming language capabilities. Traditional approaches mainly rely on metadata, an extensible set of attributes describing the Object. OO-SS explicitly separates file metadata from data to support additional capabilities and typical formats used for extracted metadata are XML, YAML and JSON. The information schema associated to an Object depends on the specific OO-SS, but, usually, it is strictly related to the features of Object itself (e.g., image size, type of compression, video duration, image resolution,...) and not to the the context where the Object has been generated.

In this paper we propose an hybrid storage system that exploit both Document- and Object-oriented storage strategies to optimize data management tasks. It is deployed into a Cloud environment able to offer a transparent storage services to the end users, which do not have knowledge of the different technologies involved, but just access data through RESTful API. Moreover, exploiting Cloud technologies means implementing a distributed and scalable service in a reliable infrastructure. We present our Cloud storage system in detail in the next section.

4. Hybrid Storage System in the Cloud

Our Cloud architecture is shown in Figure 1. It gathers data from many heterogeneous Monitoring Infrastructures (MIs) and decouples the functionalities of the Storage Systems in managing different types of data. Thus, it includes instances of both a DO-SS and OO-SS deployed in the Cloud virtual infrastructure, that are used according to well defined rules, in order to offer an hybrid storage solution efficient and versatile.

Data from MIs are collected through the *Data Gathering Interface*. It is a plug-in based interface able to interact with different information systems and communication technologies. All the collected data (both Observations and Objects) are managed by the *Data Manager*, that is in charge to abstract data, enrich data with geolocalized information, select the best storage technology for the specific type of data and, finally, insert data in the storage system. The *Identity Manager* and *Access Control* components implements security functionalities to manage users accounts and set polices to access data and services. Authorized users access data through RESTful API.

4.1. Storage System: the Data Manager

The *Data Manager* component in the Cloud architecture is responsible for collecting data coming from the MI The main functionalities of the *Data Manager* are: 1) data abstraction and 2) data enrichment.

The data abstraction task of the *Data Manager* is necessary to overcome issues related to the heterogeneity of data. It abstracts information on both monitoring devices and sensed data, providing a uniform semantic description of them. Abstracted entities interact each others and represent the real world, where things (e.g., monitoring device) observe other things (e.g., monitoring data). The Sensor Web Enablement (SWE) initiative of the Open Geospatial Consortium (OGC) has taken important early steps towards enabling web-based discovery, exchanging and processing sensing information. It defines service interfaces which enable an interoperable usage of sensor resources by defining standardized service interfaces. SWE services hides the heterogeneity of an underlying sensor network, its communication details and various hardware components, from the applications built on top of it. In this paper, we specifically refer to SWE standards to characterize data stored in the Cloud.

Even if SWE has been designed to describe Observations in a sensing environment, we adopt its semantic also to treat Objects, and to optimize querying and retrieval tasks. Indeed, traditional OO-SS rely on metadata. To fulfill monitoring purposes, it is necessary to relate the Object with the environment and, most of all, abstract information according to the SWE specifications in order to provide a seamless querying interface towards end users. To this aim,

data enrichment functionalities allow to extend the information schema of each object with context-aware metadata compliant with SWE specifications.

4.2. Data Publishing

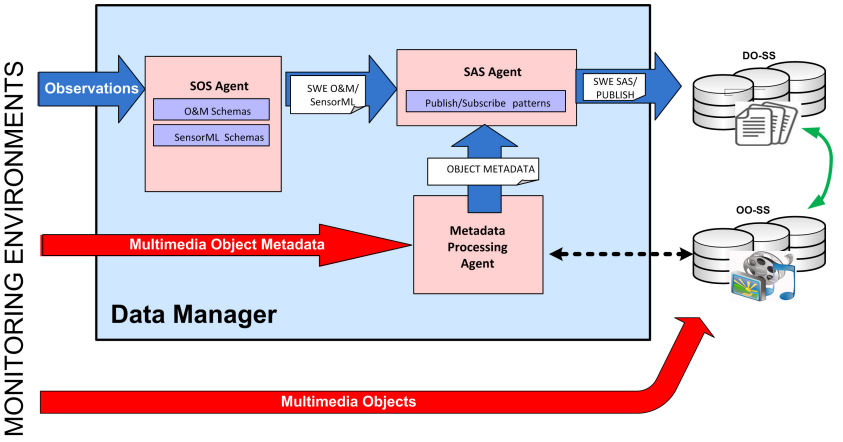

Fig. 2. Processing of data for storing

In the last year, we have widely investigated OGC-SWE specifications, especially to integrate monitoring environments in the Cloud and expose data to end-users. In this paper we focus the attention on data storage issues and, hence, we propose a new solution to organize and manage data. To this aim, we refer to two specific SWE standards, that are the Sensor Observation Service (SOS) and the SAS (Sensor Alert Service). Specifically, the SOS standard discuss interfaces for requesting, filtering and retrieving Observations and sensor system information, whereas the SAS standard describe interfaces for publishing and subscribing Observationss coming from sensors. As shown in Figure 2,the *Data Manager* contains to agents, the *SOS Agent* and the *SAS Agent*, that implements respectively the SOS and SAS specifications. As pointed by the SWE guideline, they are specifically designed to manage Observations. In particular, the *SOS Agent* supports all the functionalities for describing sensors and Observations, abstracting them in a well defined format and gathering measurements from MIs. Informations are then exposed following the specifications of the SWE SensorML and Observation and Measurements (O&M) standards. In particular, SensorML provides models and XML schemas for describing sensor systems and processes, and O&M provides models and XML schema for encoding Observations and measurements from a sensing environment.

The main task of the *SAS Agent* is to provide a platform to meet the requirements of Cloud users, which need environmental information to develop advanced services. It provides data according to the publish-subscribe model. Each type of Observation (characterized by a specific observed phenomena in a well defined MI) is identified by a *PublicationID*, and all the Observation are provided to users by publishing a *SWE-SAS Publish* document, that is an XML document including one ore more Observations related to the same *PublicationID*.

Objects can not be expressed through SWE files, but only the related metadata can be organized according to te SWE specifications to describe the content of the Object. Thus, the *Metadata Processing Agent* acts to enrich the Object with geolocalized information (e.g., time and place of acquisition, tenant, expiration time...). Such geolocalized information are provided to the *SAS Agent* that stores them ito the DO-SS. After the data enrichment process, the *Data Manager* component uploads the Object into the OO-SS. Thus, data Objects are splitted, in order to optimize the storage, querying and retrieval tasks: the metadata description is stored into the DO-SS, whereas the Object is stored into the OO-SS.

From the point of view of the end user, queries for data are always submitted to the DO-SS. Since data are related to monitoring services, queries perform gelocalized and time oriented requests. The user submits his/her request to the system and the related information is retrieved by the DO-SS. If the requested content is an Object, the retrieval process will also provide the hook to access it into the OO-SS.

5. Use case: the SIGMA Project

The Sensor Integrated System in Cloud environment for the Advanced Multi-risk Management (SIGMA) is an Italian National Operative Program (PON) project aiming to acquire, integrate and compute heterogeneous data, from various sensor networks (weather, seismic, volcanic, water, rain, car and marine traffic, environmental, etc.), in order to manage risky situation in both the industrial production process and in the territory. For example in the industry field, analyzing data coming from both several ICT equipments and the surrounding environment, it may be possible to control the production processes; considering the territory, analyzing data coming from sensors able to detect traffic congestion in a given area, it may be possible to provide useful information to the population and relevant authorities, in order to optimize routes or manage social events or natural disasters.

The SIGMA architecture has five layers. At the lowest layer there are different sensor networks. Some of them are already installed on territory, such as the SIAS network that consists of a series of weather stations to support the agriculture industry, the Water Observatory that consists of a series of hydrometric stations and rainfall to support the design of water projects, and the INGV networks for monitoring seismic and volcanic activities in Sicily, Italy. SIGMA integrates the existing networks, for multiparameter monitoring of sensitive areas and increased hydrological, hydro geological, geological, seismic, volcanic land risk, and integration with other networks such as that for car and naval traffic monitoring with GPS and GSM systems. At the second layer, the architecture holds virtualized and distributed resources provided by a Cloud computing framework. This layer is based on CLEVER, a flexible framework for inter-Cloud communications and event notification [17]. It includes specific components for virtual infrastructure set up and management, sensing environment integration and data retrieval and storage. The advantages of the framework come from the fact that it will provide computation and flexible storage capacity with enhanced performance, thus facilitating the integration of unstructured networks that make available large amounts of data to be stored and processed. At the third layer, there is the Middleware, an intermediate software layer that, through a series of interfaces, gathers data from various heterogeneous networks, standardizing them and making it available at Business Intelligence. At the forth layer, the Business Intelligence components are responsible to process data, implementing the actual business logic of the architecture. At this level, through a series of algorithms, many complex problems are solved and the results are supporting the industrial plant or territory monitoring and management activities. The highest level of architecture is finally represented by the Application layer that takes care to create interfaces for user interaction with the system (e.g. Functional Centers, Operating Rooms, etc ...).

5.1. Big Data Storage in SIGMA

The Cloud storage system presented in this paper has been impemented to fulfill the requirements of data management at the layer two-three of the SIGMA architecture. In particular, the SIGMA Cloud platform uses MongoDB as DO-SS for the storage of all information coming from the monitoring systems. MongoDB is an open source document-oriented DB, able to organize data in JSON-style documents with dynamic scheme (called MongoDB BSON documents), making the integration of data with applications easier and fast. Collections in MongoDB stores data coming from different sensor networks and monitoring environments.

To integrate the subsystem for data collection with the storage subsystem, it is necessary to use a software module (parser) which operates a fast format conversion of SWE to BSON before permanently storing such data in MongoDB. The result of this operation is a flat representation of data organized according to the logic SWE, but exposed as BSON documents.

We have implemented the OO-SS by using Swift. Swift is a widely-used and popular object storage system provided under the Apache 2 open source license. A key reason why Swift serves so well for highly-available, unstructured application data is that its design, just like Amazon S3, incorporates eventual consistency. In Swift, objects are protected by storing multiple copies of data so that if one node fails, the data can be retrieved from another node. Even if multiple nodes fail, data will remain available to the user. Swifts design for eventual consistency means that there is a guarantee that the system will eventually become consistent and have the most up-to-date version of data for all copies of the data but still provide availability to data should hardware fail. This design makes it ideal when performance and scalability are critical, particularly for massive, highly distributed infrastructures with lots of unstructured data serving global sites.

The developed storage service expose Rest API to access data. Specifically we used the Mongodb REST server written in Java and based on Jetty web server[18].

6. Conclusions

The paper deals with big data storage issues due to monitoring activities in smart environments. In particular, we have discussed what is the meaning of ”big data” in smart environment and we have identified the most suitable technologies to store different types of data. Then, we have proposed a new storage solution that integrate different types of storage technologies, that are Document and object oriented storage system, in order to optimize performance in data sotrage, querying and retrieval. The solution we proposed exploits the Cloud thus to benefit of scalability and reliability. We have provided a detailed description of our Cloud storage architecture, giving many indications on our design choices. Also, we provided some hint on the effective implementation of the storage architecture within the SIGMA project, an is an Italian National Operative Program project aimed to the monitoring of industrial production process and the territory.

Acknowledgements

The research was partially supported by the PON 2007-2013 SIGMA project and by the POR FESR Sicilia 2007-2013 SIMONE project.

The authors would like to thank Giuseppe Tricomi and Antonio Galletta, for their valuable effort in the development of the system prototype.

References

1. E.-M. Fong, W.-Y. Chung, Mobile cloud-computing-based healthcare service by noncontact ecg monitoring, Sensors 13 (12) (2013) 16451–16473.
2. X. Chen, Y. Zheng, Y. Chen, Q. Jin, W. Sun, E. Chang, W.-Y. Ma, Indoor air quality monitoring system for smart buildings, in: UbiComp 2014, ACM, 2014.
3. M. Fazio, A. Puliafito, Cloud4sens: a cloud-based architecture for sensor controlling and monitoring, Communications Magazine, IEEE 53 (3) (2015) 41–47.
4. L. Jiang, L. D. Xu, H. Cai, Z. Jiang, F. Bu, B. Xu, An iot-oriented data storage framework in cloud computing platform, Industrial Informatics, IEEE Transactions on 10 (2) (2014) 1443–1451.
5. M. Fazio, M. Paone, A. Puliafito, M. Villari, Huge amount of heterogeneous sensed data needs the cloud, in: International Multi-Conference on Systems, Signals and Devices (SSD 2012), Chemnitz, Germany, 2012.
6. C. Reed, M. Botts, J. Davidson, G. Percivall, OGC Sensor Web Enablement: Overview and High Level Architecture, IEEE Autotestcon (2007) 372–380.
7. Y. Ding, M. Neumann, D. Gordon, T. Riedel, T. Miyaki, M. Beigl, W. Zhang, L. Zhang, A platform-as-a-service for in-situ development of wireless sensor network applications, in: Networked Sensing Systems (INSS), 2012 Ninth International Conference on, 2012, pp. 1–8.
8. J. Zhou, T. Leppanen, E. Harjula, M. Ylianttila, T. Ojala, C. Yu, H. Jin, L. Yang, Cloudthings: A common architecture for integrating the internet of things with cloud computing, in: Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on, 2013, pp. 651–657.
9. S. H. Shah, F. K. Khan, W. Ali, J. Khan, A new framework to integrate wireless sensor networks with cloud computing, in: Aerospace Conference, 2013 IEEE, 2013, pp. 1–6.
10. Y. Yuan, H. Wang, D. Wang, J. Liu, On interference-aware provisioning for cloud-based big data processing, in: Quality of Service (IWQoS), 2013 IEEE/ACM 21st International Symposium on, 2013, pp. 1–6.
11. S. Rao, R. Ramakrishnan, A. Silberstein, M. Ovsiannikov, D. Reeves, Sailfish: A framework for large scale data processing, in: Proceedings of the Third ACM Symposium on Cloud Computing, SoCC ’12, ACM, New York, NY, USA, 2012, pp. 4:1–4:14.
12. M. Diaz, G. Juan, O. Lucas, A. Ryuga, Big data on the internet of things: An example for the e-health, in: Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2012), 2012, pp. 898–900.
13. O. G. Consortium, https://portal.opengeospatial.org/files/49608 (2012).
14. D. Ballari, M. Wachowicz, M. A. Manso, Metadata behind the interoperability of wireless sensor network, Sensors 9 (2009) 3635–3651.
15. T. Huang, Surveillance Video: The Biggest Big Data, Computing Now 7 (2).
16. M. Fazio, A. Puliafito, M. Villari, Iot4s: A new architecture to exploit sensing capabilities in smart cities, Int. J. Web Grid Serv. 10 (2/3) (2014) 114–138.
17. A. Celesti, F. Tusa, M. Villari, A. Puliafito., Integration of CLEVER Clouds with Third Party Software Systems Through a REST Web Service Interface, in: 17th IEEE Symposium on Computers and Communication (ISCC’12), 2012.
18. Mongodb Java REST server, https://sites.google.com/site/mongodbjavarestserver/ (2015).