© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. doi: 10.1109/TCC.2019.2908820

Big-Sensor-Cloud Infrastructure: A Holistic Prototype for Provisioning Sensors-as-a-Service

Subarna Chatterjee, *Member, IEEE*, Arijit Roy, *Student Member, IEEE*, Sanku Kumar Roy, *Student Member, IEEE*, Sudip Misra, *Senior Member, IEEE*, Manmeet Singh Bhogal, and Rachit Daga

For Personal Use Only

Abstract—The proposed work relates to the development of *Big-Sensor-Cloud Infrastructure* (*BSCI*) that immensely enhances the usability and management of the physical sensor devices. Traditional Wireless Sensor Networks (WSNs) are manufactured in a proprietary, vendor-specific design. Thus, the renderability of WSNs is almost infeasible to people/organizations that do not own a network of their own. Thus, in the existing system, WSN-based applications are inaccessible to the naive-users or common people who do not own physical sensor devices. Recently, sensor-cloud infrastructure has been viewed as a substitute for traditional WSNs. However, with the increasing growth in the velocity, variety, and variability of data, the management becomes a serious concern and difficulty. Thus, existing systems are not able to capture, analyze, and control the present data efficiently, in real-time. BSCI is a distributed framework for "Big" sensor-data storage, processing, virtualization, leveraging, and efficient remote management. The methods of the proposed BSCI are persuasive as they are equipped with the ability to handle "Big" data with enormous heterogeneous data volumes (in zettabyte) generated with tremendous velocity. The framework interfaces between the physical and cyber worlds, thereby acquiring real-time data from the physical WSNs into the cloud platform. This data are processed and delivered to the end-users as a simple service – Sensors-as-a-Service (Se-aaS). BSCI completely maintains and manages the data and the metadata internally within its database. Multiple organizations with heterogeneous demand can be successfully served with Se-aaS through BSCI. From a user-perspective, BSCI is highly convenient as the users are completely abstracted from the underlying complex processing logic. This allows the naive users to envision the typical hardware sensor devices as simple accessible services like electricity, and water.

Index Terms—Wireless Sensor Network, Sensor-cloud, Big Data, Virtualization, Prototype

◆

1 INTRODUCTION

Contemporary research has recognized sensor-cloud infrastructure as a potential substitute for traditional Wireless Sensor Networks (WSNs) [1]–[3]. Conventional WSNs possess inherently a proprietary and vendor-specific design, which are inflexible in handling application switching dynamically at runtime, due to the presence of monolithic kernels within the sensor nodes [4], [5]. Sharing of data is also non-trivial, as the WSN owners are generally unwilling to share their data to an external user in order to maintain security. Thus, in the existing system, WSN-based applications are inaccessible to the naive-users, who do not own the physical sensor devices. The limitations of WSNs led to the conceptualization of sensor-cloud [6], [7]. According to MicroStrains, one of the pioneers in this domain, sensor-cloud is formally defined as [8] – *A unique sensor data storage, visualization and remote management platform that leverages powerful cloud computing technologies to provide excellent data scalability, rapid visualization, and user programmable analysis.*

In this work, the concept of sensor-cloud is analyzed from an implementation point-of-view. The limitations of the platform are identified and a robust and holistic platform for provisioning *Sensors-as-a-Service* (*Se-aaS*) – the *Big-Sensor-Cloud Infrastructure* (*BSCI*) is proposed in this paper.

1.1 Motivation

The concept of sensor-cloud infrastructure was formally proposed by Yuriyama and Kushida [6]. The infrastructure is envisioned to support multiple end-user organizations with real-time sensor services. However, the existing sensor networks typically generate big-data with enormous *volume*, *velocity*, and *variety*, i.e., the generated data is *big* in size and are hence to be processed differently. A practical example of sensor-cloud is referred to from one of our previous related works [7] and is shown in Figure 1 for an environment monitoring application. The figure illustrates the different components involved in the functionalities of sensor-cloud specifically for the particular application. In sensor-cloud platforms, the data are handled using traditional data processing techniques, which are incapable of managing heterogeneous and voluminous data in real-time and thus affect the Quality of Information (QoI). Additionally, when multiple user-organizations are required to be simultaneously provisioned with sensor services, the existing sensor-cloud infrastructure [6], [9], [10] is likely to be overwhelmed with a huge number of data requests, thereby, creating a "bottleneck" within the sensor-cloud platform.

Subarna Chatterjee is with Inria, Rennes, France. (e-mail: subarna.chatterjee@inria.fr), but did this work during her stay at IIT Kharagpur.
Arijit Roy, Sanku Kumar Roy, and Sudip Misra are with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India. (e-mail: {arijitr, smisra}@sit.iitkgp.ernet.in, sankukumarroy@iitkgp.ac.in).
Manmeet Singh Bhogal is with the Department of Information Technology, National Institute of Technology, Durgapur, (e-mail: manmeetsinghbhogal1@gmail.com).
Rachit Daga is with the Department of Computer Science & Engineering, National Institute of Technology, Durgapur, (e-mail: rdaga92@gmail.com).

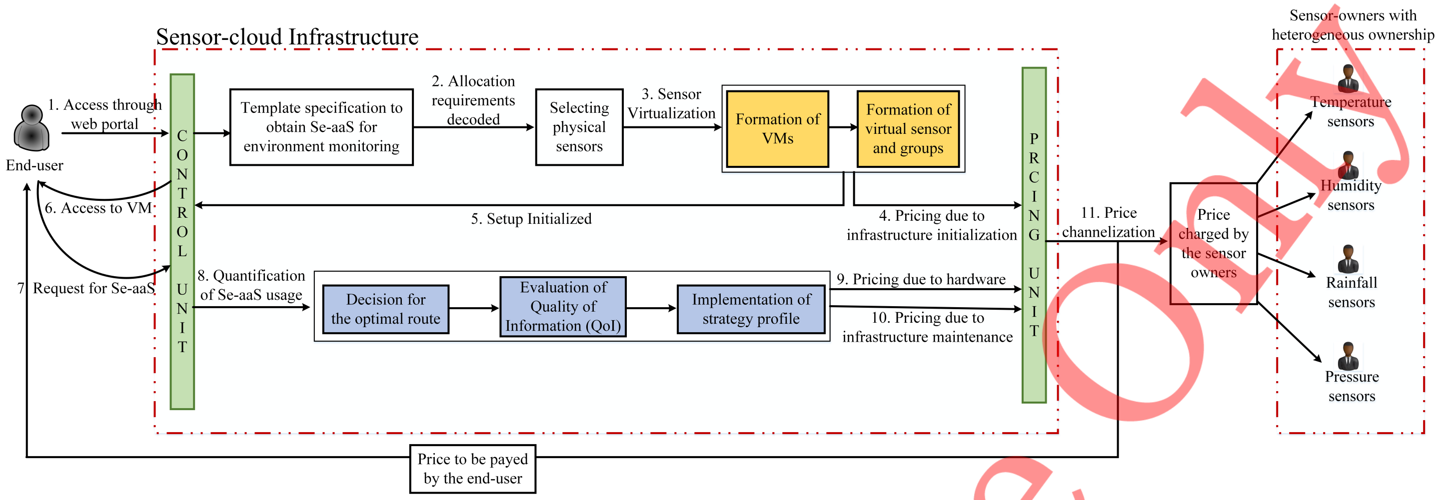


Figure 1: Operations of sensor-cloud for an environment monitoring application

Evidently, the existing sensor-cloud systems [6], [9], [10] are unable to capture, analyze, and control the present data efficiently, in real-time. Another problem that is typically encountered with the traditional systems is that, as the generated physical data is highly unstructured, the systems fail to correlate, connect, and process the huge data volumes in real-time. This becomes a real difficulty for users or organizations with a large number of queries to be processed over big-data in real-time.

As mentioned earlier, to address the aforesaid limitations of sensor-cloud infrastructure, this work proposes a new infrastructure —BSCI*. It is a platform for big-data storage, processing, leveraging, and efficient remote management. As BSCI is cloud-based, it ensures the features of scalability, pay-per-usage, and implementation of user programmable logic. The architecture allows the common people to envision Se-aaS. The underlying technology upon which Big-Sensor-Cloud thrives on is virtualization of sensor devices, modular organization of big-data, and real-time data visualization.

1.2 Contribution

The *contributions* of the work are analyzed from both technical and business perspectives. The *technical* contributions of this work are as follows:

(i) The biggest contribution of BSCI is its ability to disseminate the usability of physical sensor devices to users who do not own them.

(ii) The framework allows a single end-user organization to use multiple heterogeneous sensor devices as per demand, specific to its applications. Multiple organizations, with similar type of demand, can also be served simultaneously, at the same time.

(iii) BSCI also claims its novelty in terms of scalability of sensor-usage, managed through virtualization. Virtualization enables the end-users to enjoy a complete abstraction of the entire logic for provisioning and managing Se-aaS.

(iv) BSCI manages "big" unstructured sensor data from varied data sources and arranges, correlates, and connects the data using "sophisticated" big-data management techniques. This enables the user organizations to execute the computationally intensive queries over large data sets in less time.

(v) BSCI follows a pay-per-use pricing strategy, thereby reasonably charging the end-users only for the units of consumption of Se-aaS, exclusive of the additional overhead that would have been otherwise associated.

Table 1: A comparative study of the roles and functionality in WSN, sensor-cloud, and Big-sensor-cloud

Actors and Roles			
	WSN	Sensor-cloud	Big-sensor-cloud
Ownership	WSN-user	Sensor-owner	Sensor-owner
Deployment	WSN-user	Sensor-owner	Sensor-owner
Maintenance	WSN-user	CSP	BSCSP
Usage	WSN-user	End-user	End-user
Voluminous data	Inefficient	Bottleneck created	Can handle
Structurization of data	No	No	Yes
Query Processing Tool	SQL	SQL	Hive Query Language
Back end intelligence	Relational DB	Relational DB	Hadoop Distributed File System
Customized querying	No	No	Yes

The proposed work has its own novelty from a *business* perspective. Cloud computing technology has already proved its huge significance in practical scenarios. Sensor-cloud is also being viewed as a potential substitute of conventional WSNs [1]. With the implementation of BSCI – the proposed technology – the end-users of this technology will rapidly evolve because of its efficiency and usefulness. In this context, the work bears its own relevance as it hugely improves the restricted access of sensor networks and their resource-constrained nature. The proposed prototype con-

---

*. The authors have also filed a patent corresponding to this work which can be retrieved from the patent database at http://ipindiaservices.gov.in [11].

For Personal Use Only

tributes immensely in effective data management, storage, real-time processing, and retrieval of big sensor data. The tangibility of BSCI can be measured in terms of its ability to render Se-aaS. This positively affects the financial aspects of the end-user organizations. The *Big-Sensor-Cloud Service Provider* (*BSCSP*) also benefits from the model with the widespread dissemination and effective management of the sensor devices. To further study the differences proposed through BSCI, a comparative analysis of the various actors and the different functionalities are presented in Table 1.

The rest of the paper is organized as follows. Section 2 describes the prior work in this area. Section 3 presents the key components of BSCI. In Section 4, we discuss the high level architecture of BSCI and the flow of information within various components. In Section 5, the implementation details of BSCI are discussed. Section 6 highlights the performance analysis of BSCI and its comparison with the existing architecture. Finally, Section 7 concludes the work.

2 RELATED WORK

This Section highlights the work done in this domain so far. Cloud computing has been quite popular and emerging these days. There have been works that have focused on the various cloud services. For example, Sim [12] has proposed an agent-based cloud system in which the idea is to develop algorithms for discovery, negotiation, and composition of cloud services. Some works have proposed schemes for resource management within cloud [13], [14]. The service-oriented architecture for Software-as-a-Service (SaaS) [15], Platform-as-a-Service (PaaS) [16], and Infrastructure-as-a-Service (IaaS) [17] have also been proposed. However, the utility of cloud computing has been fully realized after it has been integrated with the physical world. After the inception of cloud computing [12], [18], research works have been initiated to integrate the traditional WSNs with the cloud platform to add sensing abilities to the cloud. Zhu *et al.* [19] have proposed the integration of sensor networks to cloud servers through trust and reputation calculation. Some of the existing works (e.g. [20], [21]) have focused on reliable data retrieval from WSNs to cloud platforms. With the growth of contemporary data in terms of three Vs characterizing big-data, i.e., *volume*, *variety*, and *velocity* [22], [23], research works have also focused on the transmission and integration of "Big" sensor data to cloud [24]. However, the concept of BSCI is not a mere integration of sensor data to cloud platforms.

In 2010, Yuriyama and Kushida [6] had formally proposed the sensor-cloud infrastructure involving virtualization of physical sensors. The work has been further extended by Yuriyama *et al.* [25], in which a service innovation architecture has been proposed to provision the services of sensor nodes. Alamri *et al.* [8] have identified the challenges of the model and Hassan *et al.* [26] have conceptualized a framework for sensor-cloud infrastructure. Eventually, Misra *et al.* [9] have proposed *sensors-as-a-service* (*Se-aaS*) and have theoretically formalized the model for sensor-cloud and justified a shift of paradigm from traditional WSNs through simulation-based analysis. The entire concept of virtualization has been further framed by Blount *et al.* [27], [28]. Different dimensions of sensor-cloud platforms such as data caching [10], data transmission [29], and developing data cloud services [30] have also been explored. Few works have analyzed the application-specific aspects within sensor-cloud such as target tracking [31] and addressing military services (Mils-Cloud) [32]. However, none of the works have focused on the development of a sensor-cloud prototype.

Motivated by the aforementioned limitations, this work is based on development of a sensor-cloud platform. In the endeavor of prototyping sensor-cloud, the loopholes in the conceptualization of sensor-cloud has become evident. With the vastness of contemporary data, especially when multiple organizations tend to access the infrastructure simultaneously with heterogeneous demand for Se-aaS, sensor-cloud encounters serious bottleneck as its processing has been traditional [6]. Sensor-cloud pursues data retrieval and extraction through traditional database management [28] and the infrastructure provisions sensor data for application-feed only, i.e., it cannot execute customized analytics on huge data sets to obtain meaningful, intelligent information. For example, sensor-cloud is able to provide answers to the query "What are the humidity measurements of city X now?", but it cannot provide the distribution of the humidity data at different times of the day or it cannot provide information on the query "Which month has the record of having the highest humidity in city X in the previous year?".

The objective of this paper is to address the aforesaid limitations of sensor-cloud and propose BSCI as an enhanced computational platform in terms of performance of data (or query) processing, storage, and management.

3 DESIGN OF BIG-SENSOR-CLOUD INFRASTRUCTURE

This Section presents and describes the design details of the proposed BSCI. Initially, the use-case diagram of the system is presented in Figure 2 in which we observe that BSCI comprises of five distinct types of actors:

(i) *End-user*: The end-users (person/organization) possess their own applications, which are to be fed with big sensor-data from the physical sensor networks. As the type and amount of the demand changes with time, the end-users enjoy scalability of Se-aaS, provided by the Big-Sensor-Cloud Service Provider (BSCSP), Thus, the end-users are privileged to demand different sensor services at different time instants from heterogeneous sensor devices, and the services are offered instantaneously by the BSCSP. In return, the end-users are liable to pay as per their usage of Se-aaS to the BSCSP.

(ii) *Sensor-owner*: The sensor-owners bear a role from a business perspective. They purchase physical sensor devices and lend them to the BSCSP. The sensor-owners earn a monthly monetary profit as per the usage of their respective sensor devices.

(iii) *Big-Sensor-Cloud administrator*: The Big-Sensor-Cloud administrator primarily manages and controls the entire cloud processing activities involving virtualization of the physical sensor devices into distinct virtual sensors, maintenance and monitoring of

For Personal Use Only

For Personal Use Only

the physical sensor devices, organization of the unstructured data, executing computationally intensive queries over the big-data sets, and real-time service provisioning of Se-aaS. However, the administrator plays a significant role in virtualization of the big sensor data, and quantifies the data usage by the individual end-users. Big sensor data segregation and filtration are also handled by the administrator.

(iv) *Big-data Controller*: The Big-data Controller operates within a Virtual Machine (VM). It is responsible for the tabular structurization and modular organization of big sensor data with each VM in a distributed manner. The controller is also responsible for the structured storage and provisioning of huge data volumes in real-time. Therefore, the Big-data Controller is a process within every VMs whereas the Big-Sensor-Cloud administrator is a software module within the cloud infrastructure.

(v) *Big-Sensor-Cloud Service Provider (BSCSP)*: The BSCSP is a business actor of BSCI. The BSCSP maintains a log of the quantified usage of the end-users, and charges price from the end-users, as per their usage of Se-aaS. It maintains a pricing policy and offers a Service Level Agreement (SLA) to the end-users.

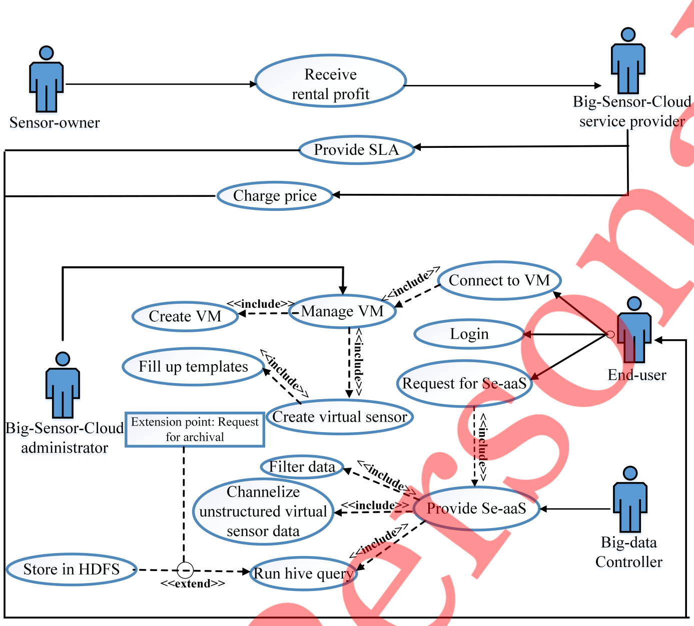


Figure 2: Use cases for Big-Sensor-Cloud Infrastructure

4 ARCHITECTURE OF BIG-SENSOR-CLOUD INFRASTRUCTURE

This Section presents the architectural details of BSCI. Primarily, it is a four-layered architecture, as shown in Figure 3. The several end-user organizations request for the sensed data to be fetched into their application from the various application-dependent physical sensor nodes. The user-organization gets connected to the Big-Sensor-Cloud service provider (BSCSP) over the client-cloud interface at the client-end (CCCI). Initially, the BSCSP provides a specific template to be filled by the end-user organization comprising of all the information relevant to the application-dependent data. These templates enable the BSCSP in mapping with the information stored in the Data-metadata repository, which in turn, helps in activating the specific physical sensor nodes spread across the physical sensor network. The raw sensed data from the various activated physical sensor nodes are transmitted to their respective nearest base stations. The raw sensed data constantly moves in large volume to the repository server of the BSCI, where they are segregated through a *Context-aware Data Filter*, based on the application dependent data. The outcome of the Data Filter is the unstructured virtual sensor data accumulating a group of specific application dependent data to be fetched into their respective requesting Virtual Machine. The unstructured data cannot be efficiently handled with the help of existing traditional technologies, because the data arrive in large volumes with huge variety and speed, thus resembling the three Vs characterizing big-data.

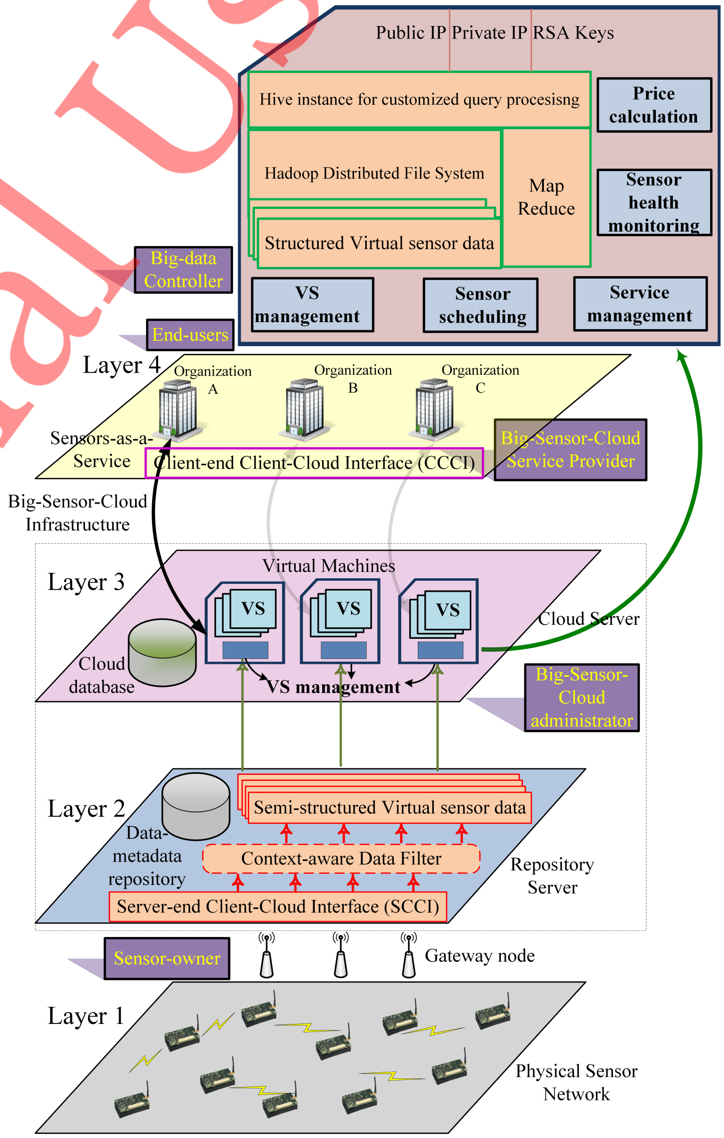


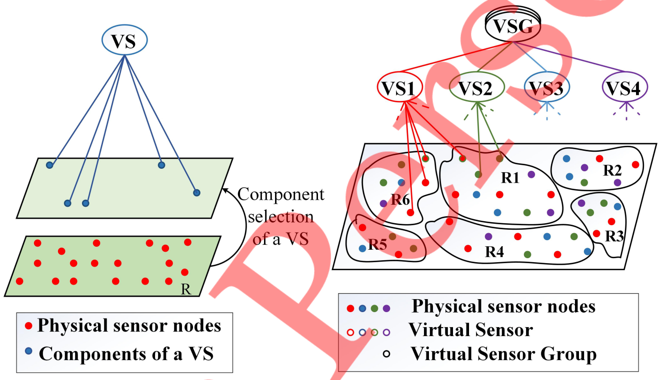
Figure 3: Architecture of Big-Sensor-cloud

Big-data within a VM can be handled using the Hadoop [33] open source software, which consists of two important layers – (a) the execution engine, known as *Map Reduce*, and (b) the file system known as *Hadoop Distributed File System* (*HDFS*). The unstructured sensor data is maintained structurally within the HDFS using the programming paradigm, Map Reduce. Map Reduce performs two basic tasks – *Map Task* and *Reduce Task*. The Map Task takes the unstructured virtual sensor data as the input, thereby producing a sequence of key-value pairs, which is sorted and shuffled by the intermediate sorting algorithm implemented between the Map and Reduce tasks. Finally, the sorted data are fed into the *Reduce Task*, which combines all the values related to a specific key and are stored within the HDFS.

Virtualization in Big-Sensor-Cloud

Having described the overall architecture and the different components of BSCI, we now present the aspect of virtualization in BSCI. Sensor virtualization is a characteristic of BSCI that enables end-users to believe and perceive that there is a dedicated sensor serving them. The diagrammatic representation of virtualization is shown in Figure 4.

Depending on the requirements of an application, an application may be served with several sensors from a particular geographical region. In Figure 4(a), multiple sensors from a particular region are grouped together to form a Virtual Sensor (VS). A VS is nothing but a logical grouping of physical sensors serving an application at a particular time [6]. To the end-user it appears that, VS is a single dedicated sensor, allocated to serve the requirements of the end-user, however, in reality, the composition of VS dynamically changes based on several factors - requirements of the application that it serves and the requirement of other applications to be served simultaneously. When multiple VSs are involved in serving an application, as shown in Figure 4(b), the logical union of VSs is called a Virtual Sensor Group (VSG). The composition of VSs and VSGs are discussed in one of our prior works [34].



(a) Formation of VS (b) Formation of VSG

Figure 4: Virtualization of the physical sensor nodes

The data from VSs and VSGs are stored and processed within VMs as shown in Figure 3. Therefore, virtualization in BSCI is a complex problem involving several aspects related to sensor scheduling, sensor allocation, sensor data management, and overall coordination. A VS or a VSG is killed only when an end-user wishes to terminate the sensor services or wishes to kill his/her application.

5 IMPLEMENTATION OF BIG-SENSOR-CLOUD INFRASTRUCTURE

This Section presents the layer-wise detailed structure of BSCI. The functional components of every layer are described.

Layer 1: Physical sensor network layer

The bottommost layer corresponds to physical wireless sensor devices that communicate with one another using the standard multi-hop routing protocols. The physical devices transmit the raw sensed data to the Big-Sensor-Cloud infrastructure through the Gateway node. It is to be noted here that, the security aspects for communication from the sensor networks can be resolved utilizing the current research on secured routing protocols [1]–[3] and is therefore not a threat from the routing perspective.

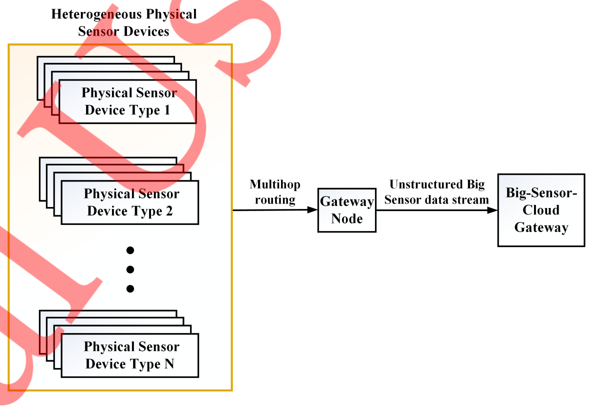


Figure 5: Block diagram of Layer 1

Layer 2: Repository Server of Big-Sensor-Cloud

The raw sensed heterogeneous data are dumped into the repository server, within which the data are further processed to generate semi-structured virtual sensor data. The individual components of the Repository Server are discussed.

2.1 **Client-Cloud Interface** (**CCI**): CCI is one of the components of Layer 2, which is further divided into *Server-side Client-Cloud Interface* (*SCCI*), and *Client-side Client-Cloud Interface* (*CCCI*). CCCI resides in Layer 4. This interface connects the end-users to the Big-Sensor-Cloud end through the user login functionality. CCI interacts with the end-users, and collects the high-level demand requirements. The requirements are interpreted in terms of resource allocation within the cloud-end by SCCI.

2.2 **Context-aware Data Filter**: The incoming big-data stream of raw sensed data is subjected to specialized filters that segregate the data as per the application demand. The filters are equipped with the ability to handle voluminous data (in zettabyte) with heavy heterogeneity generated at tremendous velocity.

2.3 **Semi-structured Virtual Sensors**: Semi-structured data formats are the ones that do not comply with the data structure format of a typical relational or

For Personal Use only

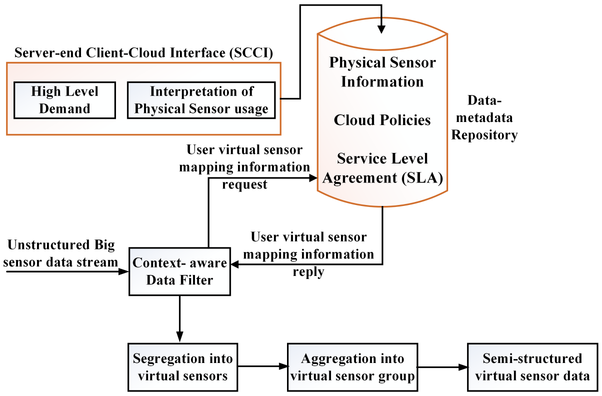


Figure 6: Block diagram of Layer 2

distributed database†. The filtered data from the sensors arrive in such formats and are grouped and aggregated into VSs and VSGs. The data within the VSs are semi-structured in nature, and are channelized to the respective VMs for further processing.

2.4 **Data-metadata Repository**: The information about the physical sensor devices and the configurations are stored in the data-metadata repository. The policies, SLAs, and the mapping of virtual sensors with the application demand are also maintained here.

Layer 3: Cloud Server of Big-Sensor-Cloud

The cloud server obtains the semi-structured virtual sensor data and routes those to the respective VMs of the respective end-users.

3.1 **Virtual Machines (VMs)**: The VMs are created dynamically based on the user-demand. The end-users connect to the respective VMs using the Public IP, and the encrypted RSA keys that are provided to them prior the connection setup phase. Once the VMs are created, the end-users obtain data from the VMs and archive within them, as per requirement.

3.1.1 **Hive Instances**: Within every VM, a Hive instance is executed for efficient processing and management of the data starting from loading of the data, to the execution of Data Definition Language (DDL), Data Manipulation Language (DML), and Data Control language (DCL) scripts.

3.1.2 **Structured Virtual Sensor Data**: The output of Hive is obtained in the form of structured virtual sensor data, which are stored with HDFS. The future queries on the big virtual-sensor data volumes are executed over the structured data sets to achieve efficiency in processing with minimum delay.

3.1.3 **Hadoop Distributed File System (HDFS)**: The result of the Hive queries are stored within HDFS from where it is transferred to the disk storage of VMs.

3.2 **Cloud database**: The cloud database stores the necessary information of the VMs, the public and private IPs of the VMs, the keys to connect with the VMs, and the mapping of the VMs with the respective end-users. The database also maintains the metadata of the structured virtual sensor information of the different VMs.

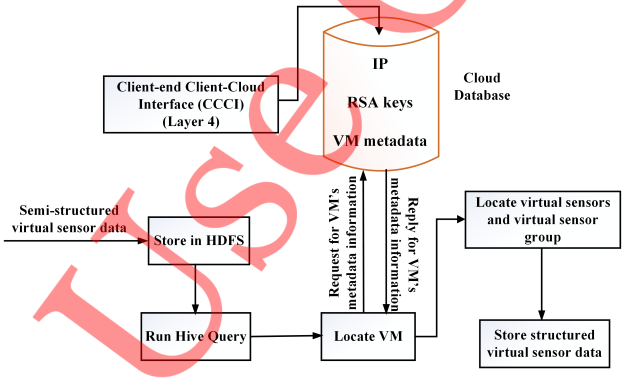


Figure 7: Block diagram of Layer 3

Layer 4: Provisioning Se-aaS to the end-users

The topmost layer of BSCI is the organizational layer, in which, multiple organizations request for Se-aaS from the BSCSP. The organizations are connected to the Big-Sensor-Cloud through the CCCI. Followed by the user login operation, the end-users are allowed to get connected to the VM and access it.

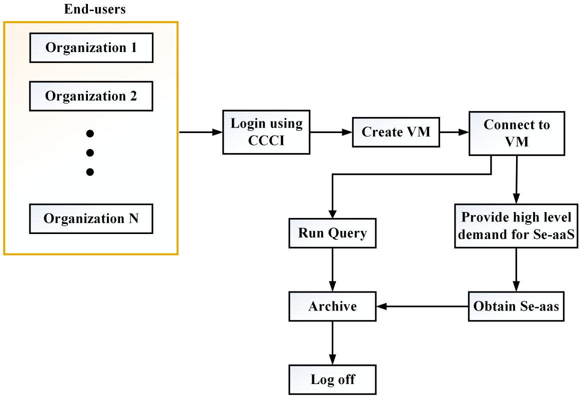


Figure 8: Block diagram of Layer 4

Customized Query Processing in BSCI

Unlike sensor-cloud, BSCI possesses the ability to accept customized queries as inputs and can process them at real-time. The query request arrives through layer 4 in which the end-users submit the queries through the CCCI. The query is immediately routed to the corresponding VMs in layer 3 that breaks the query into decodable components. The query

---

† Such data formats include Extensible Markup Language (XML), JavaScript Object Notation (JSON) notations along with data extracted from Representational State Transfer (REST) APIs or generated through OEM (Object Exchange Model) and Simple Object Access Protocol (SOAP). For storing these data, they are optimized and compressed using standard algorithms (that utilize the metadata of these data formats) and pushed to databased such as HDFS, MongoDB, Couchbase, etc.

For Personal Use Only

is decoded in terms of the physical sensor allocation and the analytics to be performed on the sensor data. Then the VMs go through a decision making process to determine if the requested data is already cached or if it can be fetched from the archives. Based on the outcome of the decision, the end-user is either fed with the cached information, or the VM initiates the process of data retrieval from the underlying sensor networks. In the latter process, data is directly fetched from the selected physical sensors in layer 1 and the BSCI converts the semi-structured sensor data into structured information in layer 2 using HDFS. The analytics are applied within the structured information and is cached within the cloud database for future references. The information is further transmitted to the end-users through the CCCI.

Table 2: Experimental setup for the underlying WSN

Parameters	**Values used**
Deployment area	500 m × 500 m
Deployment	Uniform, random
Number of sensor nodes	1000
Communication range	[50, 100] m
Transmission energy	7 nJ/bit
Computation energy	5 nJ/sec
Sensing energy	6 nJ/event
Energy due to state transition	30 nJ
Mean time between failures for a single node	5 time units

Table 3: Experimental setup for cloud

Parameters	**Values**
Processor	Intel(R) Core(TM) $i3-2105$ CPU @ 3.10 GHz
RAM	128 GB, DDR3
Disk Space	320 GB
Operating System	Ubuntu 14.04 LTS
Query Types	DDL, DML, Retrieval
No. of end-users (μ_2)	20
No. of sensor-owner (μ_1)	10
Type of sensor	5
Data size	$[5, 10, 15] \times 10^6$ record
Record size	2 byte
Time	5 years (60 months)
Nodes registered by sensor-owner (n_1)	1000
Number of nodes in the underlying WSN (n_2)	1000
Number of faulty nodes (n_3)	100 per year
Nodes used monthly on an average (n_4)	50 per user
Unit cost of a node (C_s)	20 unit
Unit cost due to deployment (C_{deploy})	3 unit
Unit cost due to maintenance ($C_{maintain}$)	10 unit/month
Unit cost due to rent (C_{rent})	10 unit/month
Cost per unit usage of Se-aaS (C_{Se-aaS})	10 unit/sensor/month

6 PERFORMANCE EVALUATION

In this Section, we analyze the performance of the proposed BSCI platform. The analysis is covered in two distinct subsections. In the first subsection, the existing sensor-cloud infrastructure [6], [9], [10] is compared to the proposed BSCI and the afore-mentioned bottleneck of sensor-cloud platforms are studied, discussed, and analyzed. Followed by this, a comparative study is also performed to investigate the sustainability of the two platforms.

The experimental data for the BSCI are collected directly by deploying the experiments on the proposed prototype. For obtaining data from sensor-cloud infrastructure, we have simulated the sensor-cloud infrastructure by reusing some of the modules from the proposed prototype of BSCI and by incorporating the implementation directives as suggested by Yuriyama and Kushida [6]. The experimental setup is identical to that of BSCI, as discussed in the subsequent subsection. For query processing and analysis within sensor-cloud, we have used traditional data processing techniques, i.e., the VMs are not equipped with distributed big data processing file systems and back-end processing.

6.1 Experimental Setup

The experimental setup for the performance evaluation of BSCI and sensor-cloud was performed on a rack server that comprised of 3 Intel(R) Core(TM) $i3-2105$ CPU @3.10 GHz processors and 128 GB of DDR3 RAM. A total of 20 end-users, 10 sensor-owners, and a single BSCSP were used for experimentation. Each VM was configured with 2 CPU cores and 4 GB of memory and a single VM was maintained for every end-user. The underlying sensor networks for both the sensor-cloud infrastructure and BSCI comprised of 1000 physical nodes out of which a maximum of 50 nodes were utilized on a monthly basis by every user. There were two categories of parameters used for the experimentation – (a) to represent the underlying WSN, as shown in Table 2 and (b) to represent the cloud infrastructure, as shown in Table 3.

A.1 Justification for the Values of Parameters Used for WSNs

The parameters that have been used to simulate sensors networks are indicated in Table 2. For analyzing the suitability of these parameters in real-life situations, we first consider the density of sensor deployment that we have used. In a deployment area of 500 m × 500 m, we have used 1000 sensors with a communication range of 50m - 100m. Therefore, it can be observed that when the WSN is initially deployed, a maximum of 8-10 sensors can optimally provide non-overlapping coverage of the network with a communication range of 100m. However, here we have certain considerations to make as follows:

- As the underlying sensor networks of BSCI comprise of heterogeneous WSNs, it is crucial to deploy several types of sensor nodes for the purpose of correctness of experimentation.
- Every node is assumed to be dead once the energy content of the node reaches zero. This means, once a node in a WSN is dead, it is important to activate other nodes of the same sensor type to ensure correctness in operations and connectivity within the network.
- Also, our work assumes that a fully-functional sensor node can undergo state transition for the purpose

For Personal Use Only

of energy conservation. Now, if a node is turned to an idle state, it implies that at that moment there is no need of its sensing hardware. However, it also implies that it cannot forward data packets of other nodes within the network. Hence, depending on the topology of the network, one or more nodes of the network may be activated to preserve connectivity.

- The communication range of the sensor nodes decrease over time. Although we begin with a small number of nodes with higher communication range, eventually, it boils down to a larger number of nodes with shorter communication range.

Based on the above considerations, we have chosen the values of our input parameters for experimentation. In addition, the choice for the values of input parameters is inspired by other related works that have performed real-life sensor deployments for practical applications [35], [36].

A.2 Justification for the Values of Parameters Used for Cloud

In BSCI, the values of the cloud-based parameters are generally expressed in units. No specific unit has been assigned. The main reason behind this is that the flexibility of scaling the magnitude of the variables is provisioned. This means that, for example, if we consider the currency unit, it can be made specific to any particular currency e.g. pound or euro or USD. In that case, the obtained results would undergo an overall change by only a multiplicative factor as a particular currency can be simply converted to another by a multiplicative factor. However, it was simultaneously important to choose the value so that the relative nature between the parameters is preserved. For this purpose, we referred to real implementation based systems [37], [38] to establish the relative dependencies within the different parameters.

6.2 Bottleneck Analysis of Existing Sensor-cloud

Figure 9 studies and analyzes the of BSCI in comparison to traditional sensor-cloud. To compare the performance of both the platforms, two different metrics are considered – average response time, and the number of queued requests.

Definition 1. *The response time ($\mathcal{R}$) of a query q_i triggered at t_i is the time difference between the instant (t_{si}) when the processing for q_i commenced and the time when it was triggered.*

$$\mathcal{R}_{q_i} = t_{si} - t_i \tag{1}$$

Therefore, for q number of queries, the average response time is obtained as, $\hat{\mathcal{R}}_q = \frac{1}{q}\sum_{i=1}^{q}\mathcal{R}_{q_i}$. To study the average response time, an experimentation is performed by varying the number of queries from 2000 to 10000 and the average response time for every test run (set of queries) is plotted by varying the data set of every query from 5 million to 20 million. Figure 9(a) reflects that with the increase in the query count and the data set size, the average response time is increased from 2.5 to 7 second averaging at 3.5 second whereas, Figure 9(b) indicates that even with large query count and data set sizes, the average response time varies from 1.2 to 2.5 second with the mean being at 1.5 second. Therefore, the response time is in general faster in BSCI than in sensor-cloud platforms.

The second experimentation is done to analyze the number of requests that may get queued up in a heavy-traffic scenario i.e., when the query count and the number of end-users increase rapidly. For a total of q application requests, the number of queued applications' (q') is given as:

$$q' = q - q_{opt} \tag{2}$$

where $q_{opt} \in \{1, q\}$ and, $\mathcal{R}_{q_{opt}} \gg \mathcal{R}_{q_{th}}$, $\mathcal{R}_{q_{opt}} > \mathcal{R}_{q_{opt}-1}$. q_{opt} is the optimal query count (in the test run) beyond which the average response time for a test run exceeds the threshold $\mathcal{R}_{q_{th}}$ and thus, creates the bottleneck.

In Figure 9(b), we observe that as the query count increases, more and more application requests are being queued and the situation is worse when the count of end-users reaches up to 10^4. It can be observed that around 500 applications are left unserved maximally in such a situation. On the contrary, in a heavy traffic scenario, BSCI maintains a moderate length of application queue varying from 18 to 38 with the mean count being close to 20.

To examine the sustainability of the two platforms, the metric of sustainability is defined as follows:

Definition 2. *Sustainability ($\mathcal{S}$) is expressed within a scale of* 0 *to* 1 *and is defined as the summation of – (a) the proportion of queries that can be responded within the average response time threshold and (b) the proportion of time spent in actually serving queries as to the time spent for both serving and queuing the queries. Therefore,*

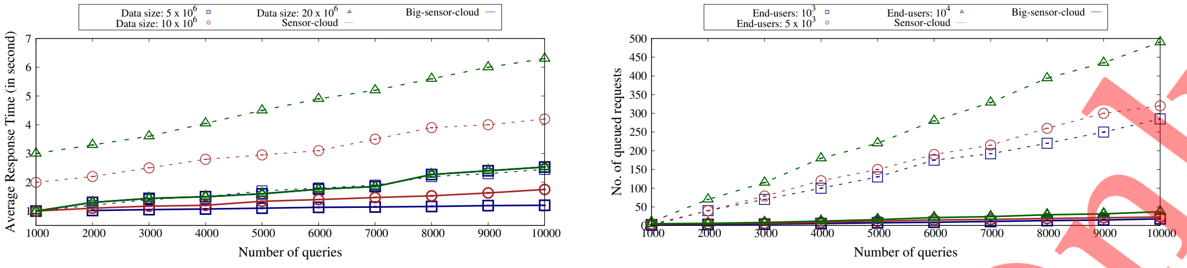
$$\mathcal{S} = \begin{cases} 1, & \hat{\mathcal{R}}_q < \mathcal{R}_{q_{th}} \\ 0.5\left(\frac{q_{opt}}{q} + \frac{\sum_{i=1}^{q_{opt}}\mathcal{R}_{q_i}}{\sum_{i=1}^{q}\mathcal{R}_{q_i}}\right), & \textit{otherwise} \end{cases} \tag{3}$$

Figure 10 reflects the sustainability of BSCI and sensor-cloud varying the two metrics – average response time, and the number of queued requests. From Figure 10(a), the sustainability reduces and eventually dies off in sensor-cloud with the increase in the average response time. However, it is better for BSCI where it sustains much longer for data size of 5 and 10 million. The sustainability reduces with large data sets of 20 million. Figure 10(b) depicts that sensor-cloud loses sustainability with the increase in the number of queued requests. However, for BSCI, the value indicated by the sustainability metric is improved and retained for a longer time.

6.3 Cost Analysis of BSCI

Here, in this subsection, we perform cash flow analysis to investigate the profitability of BSCI when compared to traditional WSNs. The data referred to the raw data transmitted from the physical sensors with a usage rate of 50±20 sensors per month by every end-user. In this case, as we mainly focused on the cash flow due to the choice of sensors and not on the complexity of analytics, we introduced a parameter of randomness so that sensors could evenly serve application requirements. We also included experimental parameters to enable an application to request for different sets of sensors

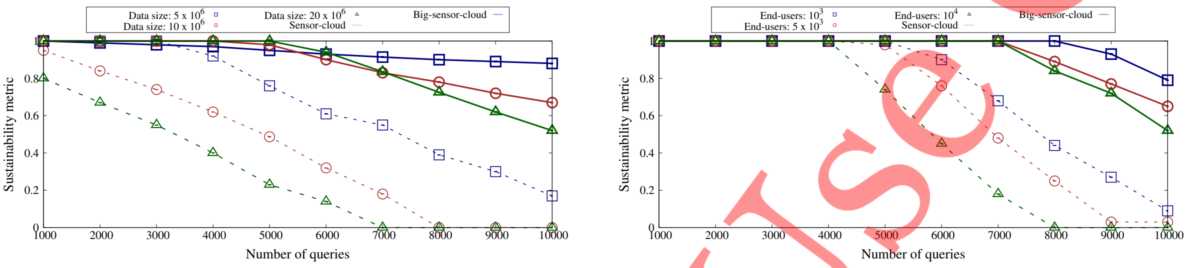
For Personal Use Only



(a) Mean response time of queries

(b) Number of requests queued

Figure 9: Comparative analysis of performance in sensor-cloud and big-sensor-cloud platforms



(a) Varied data size

(b) Varied number of end-users

Figure 10: Comparative analysis of sustainability

on a periodic basis. As obtained from our previous work [9], we model the cash flow for the actors as follows:

$$CO_{so}{}^{\ddagger} = n_1 \times (C_s + C_{deploy}); CI_{so} = n_1 \times C_{rent} \tag{4}$$

$$CO_{wsn} = \begin{cases} n_2 \times (C_s + C_{deploy} + C_{maintain}), & \text{for the } 1^{st} \text{ month} \\ C_{maintain} + \frac{n_3}{12} \times (C_s + C_{deploy}), & \text{otherwise} \end{cases} \tag{5}$$

$$CO_{end-user} = n_4 \times C_{Se-aaS} \tag{6}$$

$$CO_{bscsp} = \mu_1 \times CI_{so} + n_1 \times C_{maintain} + \frac{n_3}{12}(C_s + C_{deploy}) \tag{7}$$

$$CI_{bscsp} = \mu_2 \times CO_{end-user} \tag{8}$$

CO and CI depict the monthly cash outflow and inflow of the actors, respectively. The experimental results, as shown in Figure 11(a), demonstrate the annual cash inflow, outflow, and net profit of the Big-Sensor-Cloud Service Provider (BSCSP) for a period of 5 years. The BSCSP serves 20 end-users, and pays rental fees to 10 sensor owners. The net profit is computed by subtracting the cash outflow from the cash inflow, and is found to be positively increasing over time. Figure 11(b) shows the average cash flows for a sensor-owner with variable usage of his/her sensor devices. As shown in the Figure, the sensor-owner experiences a cash outflow only in the first year in terms of investments in procuring the sensor nodes. The outflow is nullified in the subsequent years and the inflow is based on the rental fee for the corresponding sensor usage, thereby incurring a positive net profit. In order to examine the cash flows of the end-user (in case of both normal WSNs and Big-Sensor-Cloud), the inflows are not directly measurable as such, in terms of the usage of Se-aaS. A comparative study of the cash outflow is analyzed for both the cases in Figure 11(c). The cash outflow of a WSN user access is due to several reasons — *deployment*, *maintenance*, and *overhead*, whereas the end-users of Big-Sensor-Cloud pay on a per-usage basis only for the units of Se-aaS that they consume.

6.4 Performance Analysis of BSCI

In this subsection, we perform experiments to comparatively analyze the performance of sensor-cloud and BSCI in terms of the query execution time.

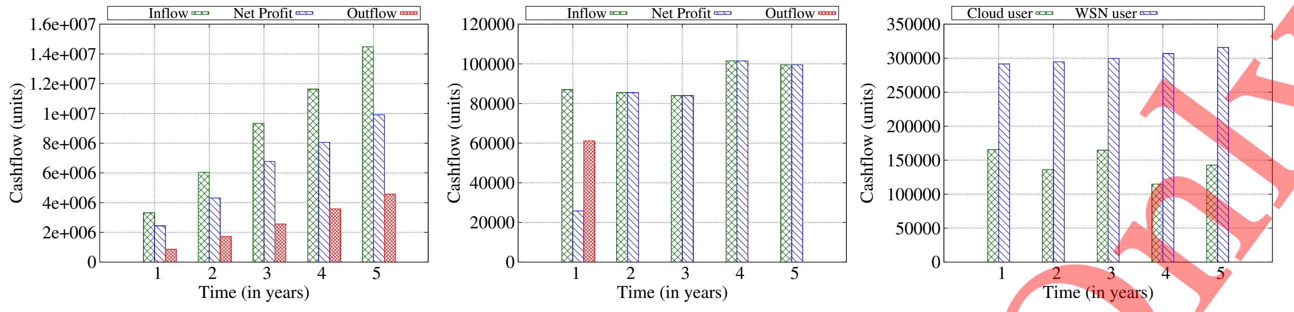
For the sake of comparison of performance of sensor-cloud and BSCI, several query types involving DML, DDL, and data retrieval are executed on varied data volumes. A comparative study is performed in terms of the execution time of the queries for sensor-cloud and BSCI. We analyse the execution time for different queries that are processed by sensor-cloud and BSCI using traditional data processing techniques and by using Hive queries on Hadoop, respectively.

Figure 12 presents a comparative study of DML query execution time for different data sets with varied size of data. Initially, experiments were performed on a data set with 5×10^6 number of data entries, as shown in Figure 12(a). For both of the paradigms, the query execution time varies insignificantly. However, the execution time is consistently low in BSCI, compared to that in sensor-cloud. This is primarily because of the fact that the entire data set undergoes through structurization of data within the Hive instance, whereas in sensor-cloud, it requires manipulation

---

‡. sensor-owner, paid only for the first month

For Personal Use Only

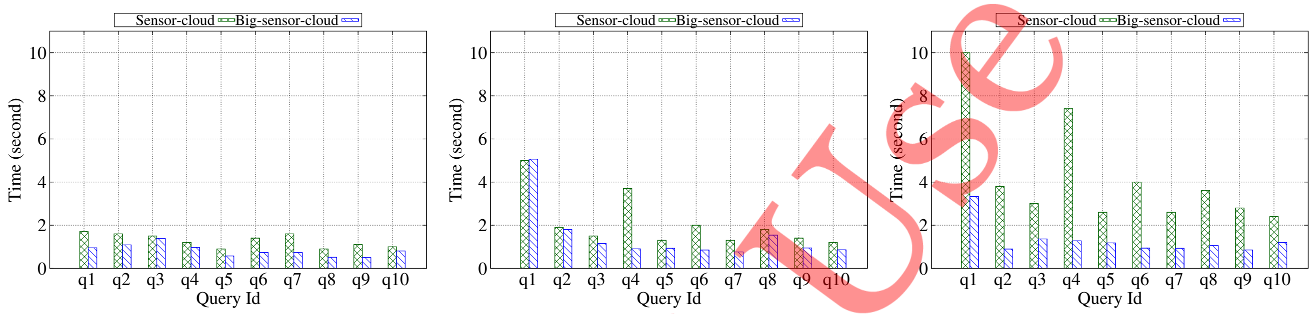


(a) Analysis for Big-Sensor-Cloud Service Provider

(b) Analysis for sensor-owners

(c) Comparison between WSN user and Big-Sensor-Cloud user

Figure 11: Cumulative cash flow analysis for the various actors of BSCI

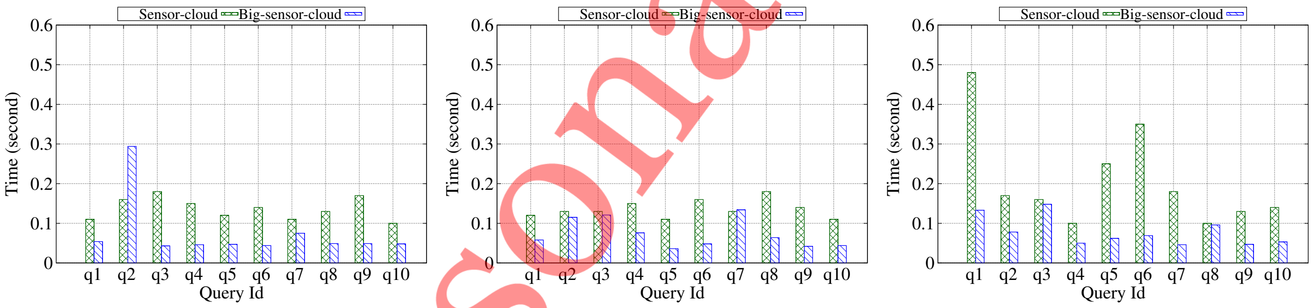


(a) Data size: 5 × 10^6

(b) Data size: 10 × 10^6

(c) Data size: 20 × 10^6

Figure 12: Analysis of DML query execution time

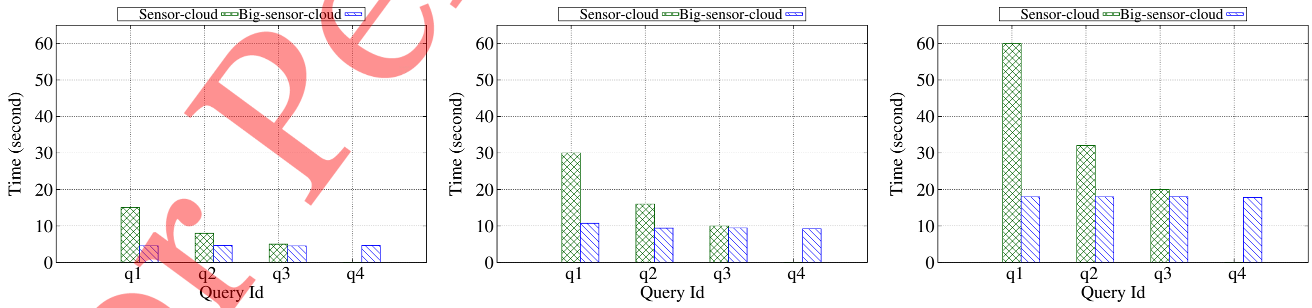


(a) Data size: 5 × 10^6

(b) Data size: 10 × 10^6

(c) Data size: 20 × 10^6

Figure 13: Analysis of DDL query execution time



(a) Data size: 5 × 10^6

(b) Data size: 10 × 10^6

(c) Data size: 20 × 10^6

Figure 14: Analysis of retrieval query execution time

of every data entry within a table of a database. For the dataset with 10×10^6 number of entries, as shown in Figure 12(b), the execution time is marginally higher for BSCI, in case of query 1 ($q1$). The principal reason behind this is the fact that

For Personal Use Only

the first query might require some additional operations for setup and configuration. For subsequent queries ($q2$ to $q10$), BSCI outperforms sensor-cloud significantly. In Figure 12(c), the results of experimentation with data sets comprising of 20×10^6 entries are shown. A huge improvement of the query execution time is observed when BSCI is used. Therefore, DML query execution time is improved using BSCI compared to sensor-cloud.

Similar experiments are performed for DDL data queries, the results of which are summarized in Figure 13. For the data-set with 5×10^6 number of entries, the query execution time is marginally lower for BSCI, as illustrated in Figure 13(a). As the number of data entries increases to 10×10^6 (see Figure 13(b)), there is substantial reduction in the query execution time. Finally, in Figure 13(c), the improvement is maximum for BSCI, compared to that in sensor-cloud.

Figure 14 highlights the analysis of query execution time for retrieval of varied data-sets. For the retrieval type of queries, the reduction in query execution time is substantially lower in most of the queries. In fact, as shown in Figure 14(c), BSCI outperforms sensor-cloud remarkably.

7 CONCLUSION

This work reports the development of a holistic prototype implementation of BSCI for realizing Se-aaS. The work addresses the problems of existing sensor-cloud infrastructure in terms of its processing ability. Unlike sensor-cloud, BSCI is a platform that handles the processing, structuring, and orientation of big-data generated from multiple organizations, simultaneously. Within each VM, a HDFS and a Hive instance are installed to enable the distributed processing of the voluminous and heterogeneous data. The work illustrates the implementation details of the infrastructure. The experimental results (for DDL, DML, and data retrieval queries) highlight the enhancement achieved through BSCI over sensor-cloud platforms in terms of the query execution time. The cash flow analysis, done for various actors, justifies the prospect of BSCI from a business perspective.

In future, large scale deployments of such infrastructure can be explored and the contextual issues of data networking, data migration, and classification of heterogeneous data induce research interest. Further, pricing of Se-aaS within BSCI can be another potential research direction.

REFERENCES

[1] M. Rapin, J. Wacker, and O. Chételat, "Two-wire bus combining full duplex body-sensor network and multilead biopotential measurements," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 1, pp. 113–122, 2018.

[2] A. Valehi and A. Razi, "Maximizing energy efficiency of cognitive wireless sensor networks with constrained age of information," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 643–654, 2017.

[3] J. Zhou, Z. Zhang, S. Tang, X. Huang, Y. Mo, and D. Z. Du, "Fault-tolerant virtual backbone in heterogeneous wireless sensor network," *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3487–3499, 2017.

[4] E. Onur, C. Ersoy, and H. Delic, "Analysis of target detection probability in randomly deployed sensor networks," *IEEE Comm. Letters*, vol. 11, no. 10, pp. 778 – 780, 2007.

[5] Z. Mao, C. Koksal, and N. Shroff, "Near optimal power and rate control of multi-hop sensor networks with energy replenishment: Basic limitations with finite energy and data storage," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 815–829, 2012.

[6] M. Yuriyama and T. Kushida, "Sensor-cloud infrastructure physical sensor management with virtualized sensors on cloud computing," in *International Conference on Network-Based Information Systems*, 2010.

[7] S. Chatterjee, R. Ladia, and S. Misra, "Dynamic optimal pricing for heterogeneous service-oriented architecture of sensor-cloud infrastructure," *IEEE Transactions on Services Computing*, 2015.

[8] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, "A survey on sensor-cloud: Architecture, applications, and approaches," *International Journal of Distributed Sensor Networks*, vol. 2013, 2012.

[9] S. Misra, S. Chatterjee, and M. S. Obaidat, "On theoretical modeling of sensor cloud: A paradigm shift from wireless sensor network," *IEEE Systems Journal*, 2015.

[10] S. Chatterjee and S. Misra, "Dynamic and adaptive data caching mechanism for virtualization within sensor-cloud," in *IEEE ANTS 2014*, December 2014.

[11] S. Chatterjee, A. Roy, S. K. Roy, S. Misra, M. S. Bhogal, and R. Daga, "Sensory network for persuasive and pervasive virtualization of physical sensors into renderable time service," Indian Patent 1145/KOL/2014, 2014.

[12] K. M. Sim, "Agent-based cloud computing," *IEEE Transactions on Services Computing*, vol. 5, no. 4, pp. 564–577, Fourth 2012.

[13] L. Mashayekhy, M. Nejad, and D. Grosu, "Cloud federations in the sky: Formation game and mechanism," *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 14–27, 2015.

[14] F. Tao, Y. Cheng, L. D. Xu, L. Zhang, and B. H. Li, "CCIoT-CMfg: Cloud computing and Internet of Things-based cloud manufacturing service system," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1435–1442, 2014.

[15] F. Liu, Z. Zhou, H. Jin, B. Li, B. Li, and H. Jiang, "On arbitrating the power-performance tradeoff in saas clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2648–2658, Oct 2014.

[16] L. Vaquero, "Educloud: PaaS versus IaaS cloud usage for an advanced computer science course," *IEEE Transactions on Education*, vol. 54, no. 4, pp. 590–598, 2011.

[17] Y. Feng, B. Li, and B. Li, "Price competition in an oligopoly market with multiple IaaS cloud providers," *IEEE Transactions on Computers*, vol. 63, no. 1, pp. 59–73, 2014.

[18] M. King, S. Platnick, W. Menzel, S. Ackerman, and P. Hubanks, "Spatial and temporal distribution of clouds observed by modis onboard the terra and aqua satellites," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 7, pp. 3826–3852, July 2013.

[19] C. Zhu, H. Nicanfar, V. Leung, and L. Yang, "An authenticated trust and reputation calculation and management system for cloud and sensor networks integration," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 118–131, 2015.

[20] C. Zhu, Z. Sheng, V. Leung, L. Shu, and L. Yang, "Toward offering more useful data reliably to mobile cloud from wireless sensor network," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 1, pp. 84–94, 2015.

[21] J. Miao, K.-P. Johnsen, S. Kern, G. Heygster, and K. Kunzi, "Signature of clouds over antarctic sea ice detected by the special sensor microwave/imager," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 5, pp. 2333–2344, 2000.

[22] X. Zhang, W. Dou, J. Pei, S. Nepal, C. Yang, C. Liu, and J. Chen, "Proximity-aware local-recoding anonymization with mapreduce for scalable big data privacy preservation in cloud," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2293–2307, 2015.

[23] L. Gu, D. Zeng, P. Li, and S. Guo, "Cost minimization for big data processing in geo-distributed data centers," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 314–323, 2014.

[24] C. Yang, C. Liu, X. Zhang, S. Nepal, and J. Chen, "A time efficient approach for detecting errors in big sensor data on cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 329–339, 2015.

[25] M. Yuriyama, T. Kushida, and M. Itakura, "A new model of accelerating service innovation with sensor-cloud infrastructure," in *2011 Annual SRII Global Conference (SRII)*, 2011, pp. 308–314.

[26] M. M. Hassan, B. Song, and E.-N. Huh, "A framework of sensor-cloud integration opportunities and challenges," in 3^{rd} *Interna-*

For Personal Use Only

tional Conference on Ubiquitous Information Management and Communication, Suwon, Korea, 15-16 Jan. 2009, pp. 618–626.

[27] M. L. Blount, M. Feridun, S. K. McIntosh, I. I. Mohomed, M. E. Nidd, A. Tanner, and B. Yang, "Sensor virtualization through cloud storage and retrieval mechanisms," US Application US 13/551,745, 2012.

[28] ——, "Sensor virtualization through cloud storage and retrieval mechanisms," US Application US 13/571,780, 2012.

[29] S. Chatterjee, S. Sarkar, and S. Misra, "Energy-efficient data transmission scheme in sensor-cloud," in 2^{nd} *International Conference on Applications and Innovations in Mobile Computing (AIMoC)*, 2015.

[30] W. He, G. Yan, and L. D. Xu, "Developing vehicular data cloud services in the iot environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1587–1595, May 2014.

[31] S. Chatterjee and S. Misra, "Target tracking using sensor-cloud: Sensor-target mapping in presence of overlapping coverage," *IEEE Communications Letters*, vol. 18, no. 8, pp. 1435 – 1438, 2014.

[32] S. Misra, A. Singh, S. Chatterjee, and M. S. Obaidat, "Mils-cloud: A sensor-cloud-based architecture for the integration of military tri-services operations and decision making," *IEEE Systems Journal*, 2014.

[33] W. Yu, Y. Wang, and X. Que, "Design and evaluation of network-levitated merge for hadoop acceleration," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 602–611, March 2014.

[34] S. Chatterjee and S. Misra, "Optimal composition of a virtual sensor for efficient virtualization within sensor-cloud," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 448–453.

[35] J. Beutel, K. Rmer, M. Ringwald, and M. Woehrle, "Deployment tech-niques for sensor networks," *International Journal on Sensor Networks Signals and Communication Technology*, pp. 219–248, 2009.

[36] M. Ringwald and K. Rmer, "Deployment of sensor networks: Problems and passive inspection," in *WISES*, 2007.

[37] N. Jain and J. Lakshmi, "Pridyn: Enabling differentiated I/O services in cloud using dynamic priorities," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 212–224, 2015.

[38] M. G. Xavier, M. V. Neves, and C. A. F. D. Rose, "A performance comparison of container-based virtualization systems for mapreduce clusters," in 22^{nd} *Euromicro International Conference on. IEEE Parallel Distributed and Network-Based Processing (PDP)*, 2014, pp. 299–306.

Dr. Subarna Chatterjee is a post-doctoral researcher at Inria, Rennes, France. She is a Google Anita Borg Scholar from the Asia Pacific region (2015) and is a Facebook GraceHopper Scholar (2016). She received her Ph.D. as a TCS Research Scholar from the Department of Computer Science & Engineering, Indian Institute of Technology, Kharagpur, India in 2017. Her current research interests include big data stream processing, cloud computing, and Internet of Things.

Arijit Roy Arijit Roy is a Council of Scientific & Industrial Research Senior Research Fellow (CSIR-SRF) and pursuing his PhD from the Indian Institute of Technology Kharagpur, India. He received his MS degree in Information Technology from Indian Institute of Technology Kharagpur, India in 2015. Prior to that, he has completed the B. Tech degree in Information Technology from the West Bengal University of Technology, India in 2010. He worked as a Network Engineer in Inteclnfonet Pvt. Ltd. Mr. Roy received several academic awards such as Smasung Innovation Award and IBM Day Award. One of his research projects was selected among the top 9 worldwide in IEEE ComSoc Student competition in 2014. The current research interests of Mr. Roy include Internet of Things, Cloud Computing, and Wireless Body Area Networks. He is student member of the IEEE and ACM.

Sanku Kumar Roy received his Bachelor degree in electronics and communication engineering from the Maulana Abul Kalam Azad University of Technology (formerly known as West Bengal University of Technology), India, in 2012. Currently, he is pursuing towards his Master of Science (by Research) degree in the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India. Along with that, he is a Junior Research Fellow in the Department of Agriculture and Food Engineering in the same institute. He has research experience in both industry and academic. Prior joining to his masters, he worked as a product development and support Engineer on the embedded system and Wireless Sensor Networking at Pervcom Consulting Pvt. Ltd., Kolkata, India. After that, he joined as Project Assistant at Indian Institute of Engineering Science and Technology Shibpur, India. His current research interests include Internet of Things, Wireless Sensor Networks, and Wireless Body Area Networks.

Dr. Sudip Misra is a Professor in the Department of Computer Science and Engineering at the Indian Institute of Technology Kharagpur. Prior to this he was associated with Cornell University (USA), Yale University (USA), Nortel Networks (Canada) and the Government of Ontario (Canada). He is the recipient of several academic awards and fellowships such as the Young Scientist Award (National Academy of Sciences, India), Young Systems Scientist Award (Systems Society of India), Young Engineers Award (Institution of Engineers, India), (Canadian) Governor General's Academic Gold Medal at Carleton University, the University Outstanding Graduate Student Award in the Doctoral level at Carleton University and the National Academy of Sciences, India – Swarna Jayanti Puraskar (Golden Jubilee Award). He has published 8 books in the areas of wireless ad hoc networks, wireless sensor networks, wireless mesh networks, communication networks and distributed systems, network reliability and fault tolerance, and information and coding theory, published by reputed publishers such as Springer, Wiley, and World Scientific.

Manmeet Singh Bhogal has pursued B.Tech. in Information Technology from National Institute of Technology, Durgapur in 2016. He has also been an intern from Indian Institute of Technology, Kharagpur, India. His current research interests include cloud computing, networking and telecommunications, and Service Oriented Architectures (SOAs).

Rachit Daga is presently working as a software engineer at Microsoft. He had interned from Indian Institute of Technology Kharagpur. His current research interests include cloud computing, big data and service oriented architectures (SOAs).

For Personal Use Only