PROCESSING IOT SENSOR DATA WITH DECENTRALIZED BIG DATA ARCHITECTURE

N.N.V.V. Praveen Kumar[1] N. Sai Nagendra[2] G.J.N.H.N. Eswar[3]

[1,2,3] Computer Science and Engineering, Pragati Engineering College (India)

ABSTRACT

IoT is an ingenious coaction of sensors and devices bulging up new challenges to security and privacy in end to end communication of smart objects. Considerations for managing the huge data injected by smart objects is a crucial part. This paper gives an overview and analysis of processing and analysing the data using Big Data techniques. Together with the applied security solutions, the paper highlights the need for a Cloud based storage methodologies in order to cope up with the immense production of sensor data. Finally, this paper proposes an intelligence framework for all the smart IoT devices.

Keywords: Sensor, Big Data, IoT, Framework, Knowledge

I. INTODUCTION

The term, Internet of Things (IoT) that refers to uniquely identifiable objects, things, and their virtual representations in an internet-like structure, was first proposed in 1998. IoT will consist of a huge number of smart devices, sensors, electronic chips, services, people and other physical objects which have potential to interconnect, interact and exchange information about themselves and the surrounding environment. Now a Days IoT has been using for regular services in day to day life. A comprehensive management framework of data that is generated by the objects should store and analysed at a high extent so as to provide seamless information to the users. A well thought out and executed Big Data and analytics strategy ultimately makes organizations smarter and more efficient. Today, Big Data is being leveraged in many industries from criminal justice to health care to real estate with powerful outcomes. Big data is a popular, but poorly defined marketing buzzword. One way of looking at big data is that it represents the large and rapidly growing volume of information that is mostly untapped by existing analytical applications and data warehousing systems. Examples of this data include high-volume sensor data and social networking information from web sites such as FaceBook and Twitter. The objective of this paper is to complement state of the art approaches by describing a comprehensive software architecture supporting the collection of sensor data produced by the IOT. In such a situation, architects must handle sensors as hardware devices, and route the produced data to data warehouses able to store the large amount of data produced by these devices. This class of architecture must tackle several challenges, e.g., data storage, avoiding processing bottlenecks, sensors heterogeneity, high throughput.

1.1 Data Sensing

Big data is not new concept or idea. However, earlier notions of big data was limited to few organizations such as Google, Yahoo, Microsoft, and European Organization for Nuclear Research (CERN). However, with recent developments in technologies such as sensors, computer hardware and the Cloud, the storage and processing power increase and the cost comes down rapidly. As a result, many sources (sensors, humans, applications) start generating data and organizations tend to store them for long time due to inexpensive storage and processing capabilities. Once that big data is stored, a number of challenges arise such as processing and analysing. Thus big data has become a buzz word in industry.

There is no clear definition for 'Big Data'. It is defined based on some of its characteristics. The big data does not mean the size. There are three characteristics that can be used to define big data, as also known as 3V's.

Volume: Volume relates to size of the data such as terabytes (TB0, petabytes(PB), zettabytes(ZB), etc.

Variety: Variety means the types of data. In addition, difference sources will produce big data such as sensors, devices, social networks, the web, mobile phones, etc. For example, data could be web logs, RFID sensor readings, unstructured social networking data, streamed video and audio.

Velocity: This means how frequently the data is generated. For example, every millisecond, second, minute, hour, day, week, month, year. Processing frequency may also differ from the user requirements. Some data need to be processed real-time and some may only be processed when needed. Typically, we can identify three main categories: occasional, frequent, and real-time.

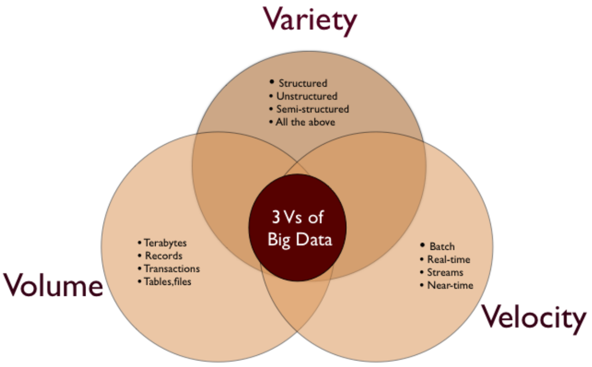


Fig.1 Characteristics of Big Data

II. EXISTING AND EMERGING FRAMEWORKS

This section outlines the current frameworks and approaches used in the Internet of Things, WSN software, Cloud Integration and Big Data. A recent survey shows that only 13 of 28 WSN systems surveyed have actually been implemented on hardware rather than run in simulators [6] and that there is still an absence of broad abstractions, which we propose later. Hence applications are often bound to a particular WSN technology and not easily portable as the application developer must have detailed knowledge of each underlying technology.

A. WSN Software Frameworks

Programming WSN applications and nodes is time consuming, error-prone and difficult requiring low level hardware and network knowledge, often using a vendor specific environment for particular hardware. Software

Engineering concepts and higher level abstractions are required to improve the development process and ease the integration with other systems in order for wider deployment of WSNs [16] as part of the seamless, context aware environments envisaged in pervasive computing [17], where applications/services are interested in the sensed information, not the underlying hardware or wireless network. Special purpose operating systems like Contiki are used on more constrained nodes, while more powerful hardware platforms such as SUNSPOT have high level language support such as Java, but at the cost of more expensive hardware and higher power consumption. TinyDB [18] essentially considers the WSN as a distributed database and can be considered limited by its table based approach and relational queries, especially in terms of handling events. Middleware approaches such as Sensation [19] treat the sensor network as a whole as an information source similar to a database, with its middleware acting as an integration layer between applications and networks and a proxy with a prior configuration for particular WSNs to hide device and network specifics. Agent based middleware requires particular node computational capability and the energy used by traffic for code mobility reduces node lifetime. A data-centric approach such as directed diffusion has the potential of significant energy savings and relatively high performance, but it is tightly coupled to a query on demand data model where applications can accept aggregated data [21]. TeenyLIME [22] is another higher level approach, which is based on a shared memory space (tuple space), derived from Linda’s [23] limited number of simple operations to insert, read, and withdraw tuples from a tuple space. TeenyLIME has been deployed in a real-world application and shown the usefulness of a tuple space approach in WSNs [24], but a node’s local tuple space is only shared with the nodes within communication range.

III. LOOP HOLES IN PRESENT ARCHITECTURE

Till now we have seen the existing architectures which have some loop holes that we need to overcome in our architecture to provide more promising and effective way to handle smart data.

3.1 Issues and Challenges

1. It is not independent of particular node hardware and it cannot handle a range of node functional capabilities and provide an extensible layered system.
2. It is not sufficient to provide abstractions for the basic operations required of a sensor node and the services using it.
3. It must provide a consistent means to exchange sensor information independent of the underlying echo and provide specific support for the modelling of sensor data to allow integration into higher level systems. A sensor node should be able to advise other nodes and services of its sensing and platform capabilities.
4. It must be able to handle small, static networks and allow the system to adapt as the network grows/changes or encounters other networks and support applications discovering and collaborating without a centralized coordination facility.
5. Data should be organized according to their category and make it robust and secure in order to handle the smart data by the organizations.

IV. PROPOSED DECENTRALIZED FRAMEWORK

1. Handling communications and provide robust knowledge extraction will be the main functionality.

2. All the sensors in smart environment will not directly communicate with each other.

3. The IoT manager will have the responsibility of taking a proper decision to forward a sensor message to another smart object.

4.Till now, the considerations for managing the huge data injected by the smart objects is neglected in IoT Research, so we embed it to the Big Data services to process the huge data for long time.

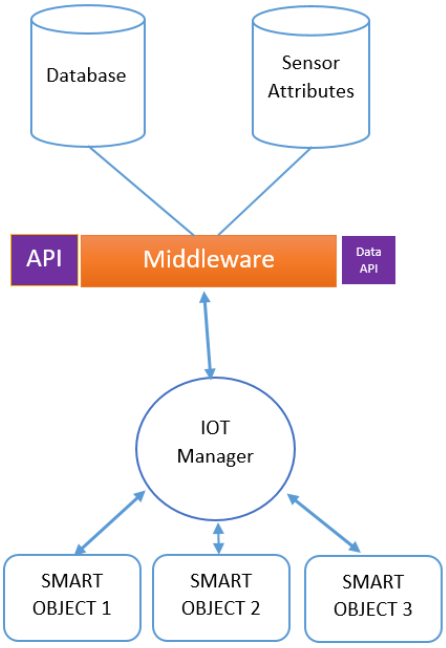
5. We propose a scheme to filter out the sensor messages which are insignificant considering some pre-specified priority levels for message transmission.

6. As we are using Big Data the raw sensor data will be process using our layered architecture and produce knowledge extraction and information visualization.

6. The IoT manager also has a fault detection, fault tolerance & fault repair mechanism.

4.1 Proposed Architecture for IoT and Big Data

Figure 2 shows the architecture for IoT and Big Data



The IoT Manager will manage the data send by various smart objects. The data will be communicated through the layers of middleware. IoT manager might be a website or an application which acts as a background framework which will manage the sensor messages which are insignificant considering some pre specified priority levels for message transmission. Consider a Smart Fire Extinguisher in a lobby which is connected through internet to the owner, works on the sensor built in it. As the temperature in the lobby increases to a certain extent the smart object will respond to it by sending information to the owner that there is a rise in temperature in the lobby and it automatically activates the fire extinguisher. What if the data send is not accurate i.e. if there is a bug in the sensor or it is hacked by the hackers, there should be an intermediate layer where the data is checked and sent to the respected owner.

Middleware: The reception middleware defines three distinct APIs: *(i)* a reception API used by the IOT Manager to send data, *(ii)* a configuration API to support the setup of measurements retrieval and *(iii)* a data API used to interact with the collected datasets. The responsibility of the middleware is to support the data reception as well as broadcasting the configuration made on the sensors to the relevant managers. The middleware contains the global sensor configuration, and the measured datasets.

4.2 Proposed Decentralized Architecture

In general, the architecture can be divided into 6 stages. Fig [3] shows the decentralized architecture for the framework.

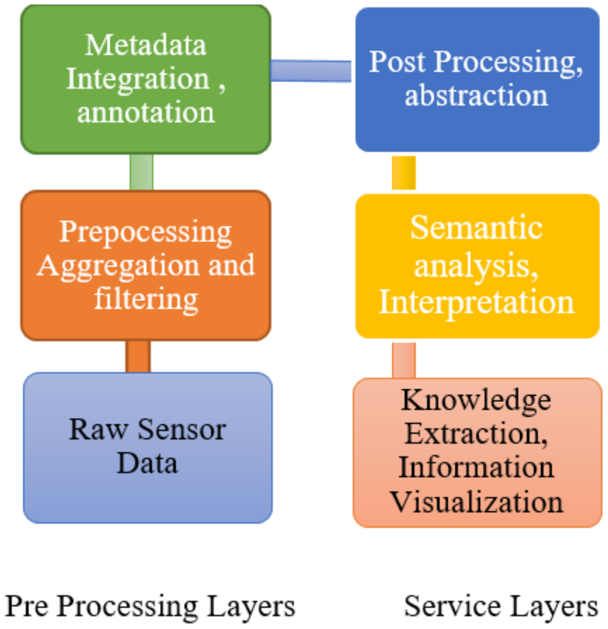

Metadata Integration , annotation
Post Processing, abstraction
Prepocessing Aggregation and filtering
Semantic analysis, Interpretation
Raw Sensor Data
Knowledge Extraction, Information Visualization
Pre Processing Layers
Service Layers

Fig 3. Decentralized Architecture for IoT and Big Data

The data production and access chain. The real-world observation and measurement data are processed and refined and/or transformed to low-level abstractions or aggregated data. Different communication, representation, publication, subscription, query, and discovery methods are then required to provide higher-level access to these data.

Sensor devices, smartphones, social media, and citizen-sensing resources are some of the key sources for producing and collecting physical-world data that can be communicated, integrated, and accessed on the Web. These resources can produce large volumes of data in which the quality of the data can also vary over time.

4.3 Advantages of the Proposed Architecture

1. Inherent cost effective due to separation of control, managing entity of the smart environment with distributed

framework.

2. The IoT Manager will have ability to synchronize the tasks of the objects.

3. Scheduling of tasks as per priority-basis can be done at IoT Manager-Level.

4. As in ubiquitous, pervasive networks, autonomy of functionalities is not a concern in our distributed intelligence-framework but the IoT Manager acts as an intelligent negotiator between the smart objects.

5. Due to the level based architecture of Big Data the data will be processed fine and data analytics is the main advantage of this architecture.

6.Data mining helps us to gather knowledge extraction and information visualization.

4.4 Threats and Challenges to Proposed Framework

The main threat to our proposed framework is the IoT manager can be compromised by false positives. The reason can be a faulty sensor or a sensor sending out signals for service even when service is not warranted. In that case the design of the application layer has to be tuned so that possibility of false positives reduces without adversely affecting the mechanism of the service layer.

As the Big Data architecture is cost effective it might be a bit less secured when compared with other cloud architectures, and if even master node fails there will be a problem with time to time synchronization.

V. FUTURE WORK

The distributed intelligence framework we have proposed does have fair degree of transparency, security but in future we propose to make it more robust. We did not develop a robust security framework and we have limited the framework with mechanisms to avoid false positives while sensors are activated. Its design principle is challenging in nature which has forced us not to consider such aspects in depth in this paper.

5.1 Possible Application Areas of Proposed Framework

The framework we propose is having enough potential for deployment in various commercial sectors like housing, healthcare, transport etc. where smart services are becoming indispensable day by day. In housing sector, a real estate group can plan smart home ventures with help of our proposed framework. In healthcare sector our framework can reduce the load on patient maintenance system by taking off some load of the system by managing data in distributed smart environments. In military operations imagine rapid and smart communication between troops our framework may connect the distributed troops and even sensors can make it possible to communicate between air force, navy, army personnel during warfare.

VI.CONCLUSION

This paper presents a novel scheme of Processing IoT sensor data with big data architecture which involves an IoT manager and a big data layered architecture which is responsible for central functionality of providing a medium of communication for the smart objects and knowledge extraction. This architecture goes from sensors to data management, and supports a user who wants to set up a research or production infrastructure to collect very large datasets in the context of the IoT.

REFERENCES

1. An Architecture to Support the Collection of Big Data in the Internet of Things, June 27 2014-July 2 2014, Pages 442-449, ISSN:2378-3818,IEEE, Anchorage Ak.
2. From Data to Actionable Knowledge: Big Data Challenges in the Web of Things, IEEE Intelligent Systems Volume:28, Issue: 6, Pages 6-11, ISSN 1541-1672, IEEE
3. A Holistic Architecture for the Internet of Things, Sensing Services and Big Data, Cluster, Cloud and Grid Computing (CC Grid), 2013 13th IEEE/ACM International Symposium, pages 546-553, Delft, IEEE.
4. Securing Internet of Things with Distributed Intelligence Framework, ICRISEM, 27 Feb 2016, ISBN: 978-81-932074-1-3, Page no:541-545
5. Hui Suo, Jiafu Wan, Caifeng Zou, Jianqi Liu, Security in the Internet of Things: A Review, 2012 International Conference on Computer Science and Electronics Engineering, 648 – 651, 978-1-4673-0689-8, INSPEC 12695467.
6. Sachin Babar, Parikshit Mahalle, Antonietta Stango, Neeli Prasad, Ramjee Prasad, Proposed Security Model and Threat Taxonomy for the Internet of Things (IoT), pp 420-429, ISSN 1865-0929, 978-3-642-14478-3.