2016 2nd International Conference on Next Generation Computing Technologies (NGCT-2016)
Dehradun, India 14-16 October 2016

IoT based Interactive Shopping Ecosystem

Abhirup Khanna
Centre for Information Technology
University of Petroleum and Energy Studies, UPES
Dehradun, India
abhirupkhanna@yahoo.com

Ravi Tomar
Centre for Information Technology
University of Petroleum and Energy Studies, UPES
Dehradun, India
rtomar@ddn.upes.ac.in

Abstract—**Internet of Things aims at integrating networked information systems to real world entities. It connects objects such as smart phones, sensors, LED(Light Emitting Diodes) displays and even clothes through the internet allowing them to interact and exchange information among themselves. In today's times IoT has found its application in practically every walk of life and inventory management is no exception. At present, inventory management is becoming increasingly complex, where there is an exponential rise in the diversity and number of both products and customers. The biggest problem faced by store owners is to optimize the experience of inventory management along with increased sales and reduced operational costs. With such enormous product lines it becomes extremely difficult for store owners to track and monitor the performance of a product in terms of its sales, shelf life, cost and customer response. IoT acts as a solution to this problem as it facilitates the use of Wireless Sensor Networks (WSN) in order to interconnect all the various actors in a logistic system. This humongous network of interconnected devices generate massive amounts of data which is difficult to store and process. Cloud computing here plays a role of a facilitator and provides great help in addressing challenges related to storage and processing capabilities. In this paper, we present an Interactive Shopping Model along with an Automated Inventory Intelligent Management System (AIMS) that benefits from the amalgamation of IoT and Cloud and provides real time monitoring, tracking and management of products. We also propose an algorithm that depicts the working of our system. The proposed system along with the algorithm are simulated using the iFogSim simulator. To this end, we illustrate the working of our proposed system along with the simulated results.**

Keywords—Internet of Things; Cloud Computing; inventory management system; iFogSim

I. INTRODUCTION

In recent times Internet of Things has gained widespread acceptance in various walks of life. According to many experts Internet of things will completely change the way people interacted with their surrounding objects thus leading to the growth of a billion dollar industry which indeed would be the new driving force for the expansion of information technology [1]. Internet of things involves connecting objects through the internet and allowing them to communicate and exchange information among themselves in order to achieve smart identification along with real time tracking, monitoring and management. IoT makes use of many information sensing technologies such as radio frequency identification (RFID), electronic label (EPC), wireless sensing technology, global positioning system, and bar code reader [2]. The concept of Internet of Things talks about an information sharing ecosystem wherein heterogeneous devices (things) are interconnected through wired or wireless networks. IoT can be considered as an advancement of machine to machine (M2M) communication which involves integration of sensors, actuators and other embedded devices in form of an IP based networking model [3]. IoT facilitates the use of Wireless Sensor Networks (WSN) in order to collect information from various sensors, which is then exchanged and depicted onto the real world through the help of actuators. IoT is not just about embedded devices connected to one another, rather, it consists of a large set of actors such as sensors, things, sensor networks, actuators and humans that lead to its proper functioning.

With such technological advancements, IoT in integration with Cloud computing finds its application in creating an automated inventory management system thus leading to the evolution of a smart shopping complex. However, since the past couple of decades many inventory management systems have been proposed but they lacked features such as real time monitoring, traceability, on the fly stock data updation, e-payment, data analytics and secure customer authentication. Traditional systems made use of manual methods or bar code technology for counting and managing logistics but failed to cope with the increasing number of customers and diversified logistics [4]. Thanks to the recent innovations that have taken place in the field of Internet of Things (IoT) and Cloud Computing which have indeed provided solutions to these problems.

Internet of things under the inventory management system creates a business information platform for optimizing and managing all activities relating to logistics thereby improving efficiency, customer satisfaction, and reduce cost. Apart from simplifying the inventory information systems, IoT also helps in shaping a smart shopping centre. It enhances the shopping experience of a customer and creates an interactive market for it. This paper aims at presenting an Automated Inventory Management System (AIMS) for optimized logistics along with an intelligent shopping complex. This automated inventory management system would optimize the inventory structure and help increase sales, reduce losses, minimize product backlog, enhance stock liquidity and prevent product wastage and deterioration. Whereas the intelligent shopping complex would allow its customers to find their desired products and purchase them without standing in long billing queues.

The rest of the paper is categorized as follows: Section II elucidates the proposed work wherein the proposed system and its working are explained. Section III discusses the algorithm that depicts the workflow of the entire system, whereas its simulation is discussed in Section IV. Finally, Section V concludes the paper.

II. Proposed Work

In the past, many inventory management systems have been proposed but most of them had their focus on the source and destination rather than on the journey. In this section, we present an Interactive Shopping Model along with an Automated Inventory Management System (AIMS) that is based on the amalgamation of Internet of Things and Cloud computing and provides efficient access, tracking, ordering, monitoring and purchasing of a product with minimal human involvement [5]. The system would provide real time asset monitoring, management and end to end traceability of the entire logistics chain. The aim of this system would be to create an interactive shopping ecosystem wherein a customer need not waste time in searching for a particular product or standing in long billing queues. The system would act as a guide for the customer letting him know where his desired products are located. In totality the system would be functioning in an automated manner thus ensuring an efficient logistic chain, quality control, cost benefits and customer satisfaction.

A. Infrustruture Layer

In present times every person is aware of the term supermarket or shopping mall. These are places which provide wide range of products such as food items, grocery, dairy products, household products, baked goods, clothing, shoes, etc. But whenever one is reminded of such a place the first thing that strikes are the hundreds of people searching for their respective products along with the never ending billing queues. The hustle of searching a product at various stores located on different floors is not a new problem. With the aim of addressing this problem, we propose an interactive shopping model where the market place itself would be informing the customer about the location and prices of the products which it wishes to buy. In this sub section we would be discussing about all the actors (physical entities) which constitute the system, their roles, basic functionalities and the way in which they communicate among themselves. The following figure denotes the workflow of our proposed system.

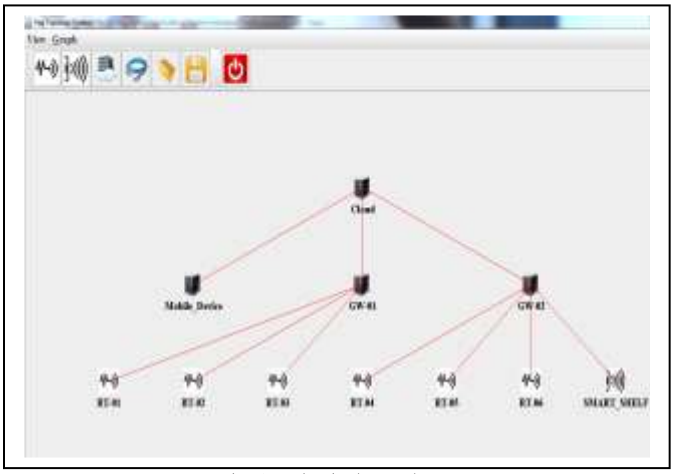

Fig. 1. Physical Topology

- *Sensors*: Sensors are the eyes and ears of the system as they detect occurrence of events, surrounding conditions and transmit the collected information. The work of the sensors is to monitor and perceive events or phenomena that take place in the physical world. Every sensor can be categorized on the basis of three parameters namely, sensor type, methodology and sensing parameters. Sensor type defines which type of sensor it is i.e. whether it is a homogeneous or a heterogeneous senor or it is a single dimensional or a multidimensional sensor. Methodology talks about the ways in which a sensor gathers information. It can be either active or passive in nature. Sensing parameters are the number of parameters which a sensor can sense. A sensor might just sense one parameter like body temperature or many parameters like in the case of an ECG. In our case we would be using RFID tags and readers for sensing information. For our system we make use of passive RFID tags as they can be easily embedded in products along with being cost effective.

- *Gateway*: All the information that has been sensed and collected by the sensors is transmitted to the gateways. Gateways act as a common point of contact wherein diverse kinds of information coming from heterogeneous types of sensors gets collected. It is the gateway which is responsible for the global addressing of each store by making use of IPv4 addresses. Each gateway is allotted a coverage area, wherein each store has been given an IP address thus facilitating efficient identification of products being sold at that store. Every gateway is allocated more than one store so as to enhance the granularity of store identification. The gateway also keeps track of its neighboring gateways along with the total number of stores functioning in its area. Finally the gateway transmits all forms of unstructured information to the subsequent cloud servers.

- *Cloud*: It acts as the storage and processing unit for the entire system. All requests coming from the customer are addressed at the cloud end [6]. It also keeps track of all the stores located in a shopping complex along with the details of every product being available at respective stores. It maintains database where information pertaining to every product such as product ID, product name, date of manufacture, name of manufacture, cost of the product, any special discount being offered on that product, shelf life, payment info and its RFID tag serial number. During the process of purchasing a product, scanned serial numbers are sent to the cloud along with their payment details. Once the purchase has been made RFID readers are informed about it and asked to update the product status of respective products.

- *Mobile Device*: It acts as a representative for a customer. It is the mobile device which allows the customer to search and purchase products on a real time basis. Any notification or special offer pertaining to a particular product is directly communicated to the mobile device in wait of an appropriate response.

- *Smart Shelf*: These are special shelves that comprise of RFID readers. Each shelf of a store may have three to five RFID readers installed on it. The work of these readers is to sense the presence of a product and read and write specific information on the RFID tags embedded on them.

B. Application Layer

In this subsection, we would be discussing the application layer of our Interactive Shopping Model. This layer comprises of four different application modules which constitute the working and the information flow of the entire system. We also specify the kind of information which each of these modules process and communicate. These three application modules have also been mentioned in the Section IV.

- *Controller*: The purpose of the controller is to manage and govern the entire system. It is the controller which orchestrates the functionalities of other application modules and entities within the system. The controller resides at the Cloud end and has detailed information regarding every product, customer, supplier, source, destination, sensor node, actuator, gateway and proxy server. All of this information is stored and processed by the controller in order to generate optimized sales preferences and logical inventory management. It is the controller which performs all calculations relating to various stocks of different products are performed. Inventory levels are maintained and managed by the controller. Item performance is one such functionality of the controller which is done in order to determine the profitability of a product. The process involves tracking of a product in terms of its sales, price fluctuation, discounts offered and customer feedback over a stimulated period of time. This process is conducted for every product in an automated manner without any human intervention. The results of this process have a great impact on the stock levels of a particular product as the system makes re-orders on the basis of item performance. Effective sales strategies are developed at the cloud end based on the purchase history of every customer. Depending upon this information suitable discount offers and promotional notifications are send to every customer. The controller establishes a one to one connection with the gateways and circulates all of its orders through it. It is the controller which generates detailed stock and inventory supply reports on the basis of product sales, inventory budget, quality control, stock levels, stock movements, item performance and customer feedback.

- *Mobile application*: It resides on the mobile device of a customer and allows them to search and purchase products. Every product has been attached by an RFID tag which comprises of information such as serial number, product ID, cost and product status. In case a customer wishes to buy certain set of products he then first needs to scan the serial number mentioned on every product and then make suitable payments through his mobile application according to the cost of the product.

- *Reader application*: It is responsible for performing read write operations on the RFID tag of a particular product. It receives signals from the controller in order to update and alter information pertaining to a product. Once the product has been purchased it is the work of the reader application to update its status from being available to sold. It is the reader application which ensures that no unpurchased product is taken outside the store and if done thus results in ringing of the store alarms to in order to notify the respective authorities.

III. ALGORITHM

The working of the AIMS could be explained through the illustration of the algorithm that forms the core for it. The algorithm depicts the workflow of the system by representing the relationship between various actors and the information that they share in form of parameters among themselves.

2016 2nd International Conference on Next Generation Computing Technologies (NGCT-2016)
Dehradun, India 14-16 October 2016

TABLE I. SYMBOL TABLE

Symbol	Meaning
α	Customer ID
ß	Password
Φ	Product Cost
γ	Brand Name
Ψ	Price Range
δ	Time
θ	Product Type
β	Product ID
St	Shop Name
SN	Serial Number
PS	Product Status (0 for available 1 for sold)

TABLE II. FUNCTION TABLE

Function	Functionality
Start()	It marks the start of the entire system
Controller()	It orchestrates the flow of information and controls the working of the entire system.
Product_Info()	It is used to keep track of all the information regarding a particular product such as Product Code, Product Price, Product Name, Price, Expiry date of the Product, etc.
Mobile_Application()	This function represents the mobile application. All queries relating to searching or purchasing of a product are carried out through this function.
Payment_Portal()	This function keeps track of all the information relating to the payments made with respect to the purchases made by the customer. Payments can be made through Credit/Debit cards or e-wallets.
Tag()	It keeps track of all the information related to an RFID tag such as serial number, product status and product ID.
Alert_Theft()	This function prevents theft of any item form the store. The function would trigger an alert in case any product is moved outside the store without being purchased.
Stock_Mgmt()	It maintains all the information pertaining to the stock of a particular product.
Order_Level()	It maintains the purchase and dispatch orders with respect to a product.
Investment_Ratio()	It calculates the investment ratio along with the inventory budget based upon various logistic parameters.
Stop()	It marks the end of successful working of the entire system

Step 1: Start()
Step 2: Controller()
Step 3: Product_Info()
Step 4: Mobile_ Application()

Initialization Block

Step 5: (αi, βi) → Mobile_ Application ()
Step 6: Mobile_ Application (αi, βi, δ) → Controller ()
Step 7: Mobile_ Application (θ, γ, Ψ, δ) → Controller()
Step 8: Controller (Si, θ, γ) → Mobile_ Application()

Step 9: Tag (SNi) → Mobile_ Application()
Step 10: Mobile_ Application (Φi,βi) → Payment_Portal()
Step 11: Payment_Portal (βi, αi, δ, PI) → Controller()

Step 12: Controller (SNi, βi, δ, PSi) → Reader()
Step 13: Reader (SNi, βi, δ, PSi) → { Tag(), Alert_Theft() }
Step 14: Alert_Theft(ω) → Product_Info()
Step 15: Controller (PSi, βi, δ) → Product_Info()
Step 16: Controller (PSi, βi, δ) → Stock_Mgmt()
Step 17: Stock_Mgmt() → Investment_Ratio(), Order_level()

Step 18: Stop()
Step 19: End

Step 1 to Step 4 constitute the initialization block. The functions being called in them are the first ones to be executed once the system gets started. Step 5 to Step 8 depict the process in which a customer searches for a certain set of products and how the system responds to these queries. Step 9 to Step 11 are performed in case a customer finalizes a product and makes suitable payments with regard to it. Step 12 to Step

14 illustrates the process in which a product has been purchased and its status gets changed from available to sold. The anti theft readers are also informed about this change in product status and as a result do not trigger alerts when the product is moved out of the store. Step 15 and Step 17 represent the automated nature of inventory management as information regarding sales and quantity of products sold and available are processed at the cloud end. This information processing leads to many useful insights such as maximum, minimum, re-order stock level, Economic Order Quantity, Inventory Turnover Ratio and an optimized and customized inventory budget.

IV. SIMULATION & RESULTS

The above mentioned algorithm is implemented on iFogSim framework. In iFogSim [7] there are various predefined classes that provide a simulation environment for Internet of Things combined with the benefits of cloud computing. It enables the simulation of resource management and application scheduling policies across edge and cloud resources under different scenarios and conditions. iFogSim is a java based simulation toolkit and can be implemented either using Eclipse or NetBeans IDE. In our case we would be using the eclipse IDE. To run iFogSim on eclipse, we first need to download the eclipse IDE and install it. After successful installation of eclipse IDE, download the latest iFogSim package, extract it and import it in eclipse. Talking of our proposed work we have created our own classes in iFogSim and have portrayed our algorithm in form of java code. The following are the screenshots that depict the working of our algorithm on iFogSim framework.

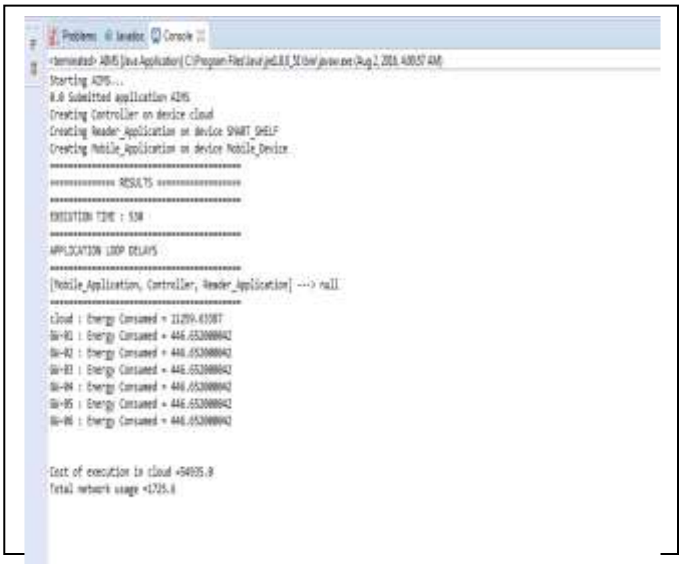

Fig. 2. AIMS Simulation

The above screenshot depicts the working of the Automated Inventory Management System. It shows how various application modules are created and allowed to run on different physical infrastructures. First of all the physical entities have been created programmatically and then allowed to run and communicate among themselves. There are in total five physical entities that have been created which have been listed in table x along with their respective configurations. Some pre-defined functions such as createFogDevice(), Sensor(), Actuator() and addAppEdge() from iFogSim have been used extensively throughout this experiment for creating physical entities and establishing connections between them. Certain user defined functions namely, add_Mobile_Device() and add_Gateway() have been used for creating mobile devices and gateways respectively. There are three application modules namely (controller, mobile_application and reader_application) that have been created for transfer and processing of information. There are certain data dependencies between these modules, which are modeled using the AppEdge class in iFogSim.

TABLE III. ENTITY TABLE

Physical Entity	Processing Capability (MIPS)	RAM (MB)	Uplink Bandwidth	Downlink Bandwidth	Level
Cloud	44000	40000	100	10000	0
Gateway	2000	4000	10000	10000	1
Smart Shelf	100	1000	10000	10000	2
Mobile Device	500	2000	10000	2500	1
RFID Tag	NA	0.0064	10000	10000	2

In order to make our experiment more realistic we had specified the end-to-end latency of every device loop. Following is the table that lists network latencies between various devices.

TABLE IV. LATENCY TABLE

Source	Destination	Latency (milliseconds)
Mobile Device	Cloud	100
Gateway	Cloud	80
Smart Shelf	Gateway	6
RFID Tag	Smart Shelf	2
Gateway	RFID Tag	4

During the simulation of our proposed system various metrics reported by iFogSim were collected. The results of the simulation are demonstrated using three different parameters namely, execution time, energy consumed (Mega Joules) by different category of devices and total network usage (KiloBytes).

V. CONCLUSION

Inventory management is one of the many domains where IoT finds its application. The paper intends to put forward an Automated Inventory Management System which is a significant step towards creating a well optimized logistics chain along with an interactive shopping market. The system that we present is based on the amalgamation of Internet of Things and Cloud computing and provides a real time view of the entire logistics process thereby improving the efficiency of inventory operation by saving human resource and reduced operational cost. The paper also talks about, the physical topology as well as the application model for our Automated Inventory Management System.

REFERENCES

[1] Luo, Z., & Wang, H. (2012, May). Research on intelligent supermarket architecture based on the internet of things technology. In *Natural Computation (ICNC), 2012 Eighth International Conference on* (pp. 1219-1223). IEEE.

[2] Li, R., & Luo, H. ZhigangBao. Based on the Internet of Things the Supermarket Chain Management Information System Development and Safety Stock Research. In *2010 2nd International Conference on Education Technology and Computer (ICETC)*.

[3] Khanna, A. (2015). AN ARCHITECTURAL DESIGN FOR CLOUD OF THINGS. Facta Universitatis, Series: Electronics and Energetics, 29(3),357-365.

[4] Mhlaba, A., & Masinde, M. (2015, July). Implementation of middleware for Internet of Things in asset tracking applications: In-lining approach. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)* (pp. 460-469). IEEE.

[5] Wang, M., Tan, J., & Li, Y. (2015, June). Design and implementation of enterprise asset management system based on IOT technology. In*Communication Software and Networks (ICCSN), 2015 IEEE International Conference on* (pp. 384-388). IEEE.

[6] Khanna, A. (2015, September). RAS: A novel approach for dynamic resource allocation. In Next Generation Computing Technologies (NGCT), 2015 1st International Conference on (pp. 25-29). IEEE.

[7] Gupta, H., Dastjerdi, A. V., Ghosh, S. K., & Buyya, R. (2016). iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments. *arXiv preprint arXiv:1606.02007*.