Sensor Data Repository System for Mobile Cloud

M.Chithik Raja

Department of Information Technology, Salalah College of Technology, Oman

Abstract

In this paper, we explained a sensor data repository system in Mobile Cloud. The development of wireless technologies, such as 3G and Wi-Fi, and the rapid growth of mobile devices equipped with sensors have enabled the practical use of Mobile Participatory Sensing (MPS). By gathering and utilizing sensor data using mobile devices, the deployment cost of services can be reduced. In the context of MPS, it is important to establish a method of storing and locating sensor data collected by millions of mobile devices. In this paper, the development of a sensor data repository system for a large-scale Mobile participatory sensing platform is proposed. By storing sensor information in the mobile device’s storage, the storage cost can be distributed. The proposed method of tracking the acquisition locations of sensor data can reduce management costs. In addition, a cache mechanism that can minimize duplicate transmissions of sensor data from mobile devices due to overlapping queries is introduced. Based on four day simulation, the proposed method can reduce the management cost of the acquisition locations by 90%. Furthermore, the cache method can reduce the transmission of duplicated sensor data on mobile devices.

Keywords: Mobile Participatory Sensing, Cloud Computing, Mobile device, MAaaS, P2P, MCE.

1. Introduction

Currently, WSNs are being utilized in several areas like healthcare, defence such as military target tracking and surveillance government and environmental services like natural disaster relief, hazardous environment exploration, and seismic sensing, and so forth. These sensors may provide various useful data when they are closely attached to each of their respective applications and services directly. However, sensor networks have to face many issues and challenges regarding their communications (like short communication range, security and privacy, reliability, mobility, etc.) and resources (like power considerations, storage capacity, processing capabilities, bandwidth availability, etc.). Besides, WSN has its own resource and design constraints. Design constraints are application specific and dependent on monitored environment. Based on the monitored environment, network size in WSN varies. For monitoring a small area, fewer nodes are required to form a network whereas the coverage of a very large area requires a huge number of sensor nodes. For monitoring large environment, there is limited communication between nodes due to obstructions into the environment, which in turn affects the overall network topology (or connectivity). All these limitations on sensor networks would probably impede the service performance and quality. In the midst of these issues, the emergence of cloud computing is seen as a remedy.

Sensor-Cloud infrastructure has been evolved and proposed by several IT people in the present days. Sensor-Cloud infrastructure is the extended form of cloud computing to manage the sensors which are scattered throughout the network (WSN). Due to the increasing demand of sensor network applications and their support in cloud computing for a number of services, Sensor-Cloud service architecture is introduced as an integration of cloud computing into the WSN to innovate a number of other new services. When WSN is integrated with cloud computing environment, several shortfalls of WSN like storage capacity of the data collected on sensor nodes and processing of these data together would become much easier. Since cloud computing provides a vast storage capacity and processing capabilities, it enables collecting the huge amount of sensor data by linking the WSN and cloud through the gateways on both sides, that is, sensor gateway and cloud gateway. Sensor gateway collects information from the sensor nodes of WSN, compresses it, and transmits it back to the cloud gateway which in turn decompresses it and stores it in the cloud storage server, which is sufficiently large. Sensor-Cloud can be used in many real-life applications like environmental monitoring, disaster monitoring, telemetric, agriculture, irrigation, healthcare, and so forth. As an illustration, we can use the Sensor-Cloud infrastructure for deploying health-related applications such as monitoring patients with cardiovascular disease, blood sugar follow-up, sleep activity pattern monitoring, diabetics monitoring, and so forth. In traditional approach, the trials of individual’s data like level of blood sugar, weight, heart rate, pulse rate, and so forth are reported everyday through some telemedicine interface. The patient’s trial information is sent to a dedicated server and is stored there for doctors or caregivers to analyse it sometime later. This system suffers from a level of adversity when the patient randomly moves from its current location, that is, when a patient is “on the go.” Thus, a more progressive, rapid, and mobile approach is needed where the recorded data from several sensor nodes of a WSN can be processed in pipelined and parallel fashion, and thereby to make the system easier to scale and be cost-effective in terms of resources available. The pipeline processing of data sets or instructions enable the overlapped operations into a conceptual pipe with all the stages of pipes processing.
simultaneously but handling of the sensor data stream is not that straightforward and will be dependent on the nature of the algorithm. The integration of Sensor-Cloud can serve as a remedy in this direction. Due to the rapid growth of mobile devices equipped with sensors, Participatory Sensing has received widespread attention in the field of sensing. In Participatory Sensing, services gather information by using mobile devices owned by individuals. Service providers can reduce costs by using resources owned and obtained by mobile device users. In addition, by combining the sensor information with geographical positions, the practical use of the sensor information by searching with geographical locations is possible. The storage cost of the concentrated central storage server is problematic when the server collects sensor information on a daily basis and for a long time. The rapid growth in the sales volume of mobile devices is expected to reach 40% in the year 2015 by the Indian government. We propose Mobile Participatory Sensing Data Store (MPS-Data Store), a distributed storage of mitigating the increase of storage cost due to the growth of mobile device users. MPS Data Store stores sensor information in the mobile devices storage. When service providers send a query to get the sensor information, MPS Data Store transfers the query to the mobile devices which have stored the designated sensor information. Thus, the proposed system can scale-out storage by storing sensor information in mobile devices. In addition, mobile devices send sensor information when and only when service providers require the information. MPS Data Store can reduce the management cost of the acquisition locations while it enables geographical range searches for retrieving sensor information. Furthermore, by using the cache function, MPS Data Store can reduce the size of sensor information transmitted by mobile devices.

2. BACKGROUND AND MOTIVATION
Mobile Application as a Service (MAaaS) In the MAaaS cloud delivery model, the cloud acts as the host and the medium through which all the mobile applications can be accessed. Connectivity to the cloud is provided through a thin mobile client installed on the mobile device. On the other hand, mobile applications hosted on the cloud are accessed through web wide gets, i.e. applications which can be installed and used within a web page, thus making them platform independent. Furthermore, the cloud ensures efficient collection, distribution, processing and delivery of the data required by mobile applications. Developers can use the infrastructure as a services and platform as a service cloud delivery models.

People-Centric Applications several cloud-based applications exploited mobile devices in a people-centric fashion. In biketastic, VTrack and mobile millennium project, network based localization and GPS input from mobile devices were exploited to determine the location of users so as to develop a traffic monitoring and navigation system. In the pier project, GPS data from end users were linked to the available information about the visited location in order to provide a report of environmental impact based on the observed mobility pattern. In the Garbage watch, Urban Atmospheres and CenceMe projects, users were recruited to actively provide sensory data – such as location, video and audio recordings from the camera and the microphone of the mobile device, as well as textual data- for environmental monitoring and for improving people’s lifestyle. In GPS and gyroscope data from mobile phones were exploited to discover potholes on roads and alert travellers. In a cloud-based social networking site was developed to facilitate information flow for aid and search as well as damage assessment, enumeration and collaboration in disaster scenarios.

Need for an Integrated Framework As apparent from the previous discussion, several applications share some needs as for the collected sensory data. For instance, location is required by almost all the above mentioned application, while gyroscope readings are needed in and different applications running on a single mobile device and capturing data individually are not the most efficient choice for the mobile device, in terms of both resource usage and energy consumption. In architecture for massive urban scanning using vehicular and mobile sensors was presented for reliable data gathering and storage. However the issue of integrated sensing was not addressed. Essentially, for each application an individual data flow is maintained in the cloud, which increases the stress on the cloud infrastructure when dealing with a large number of users. We overcome this drawback with the help of a unified sensing and data delivery model to the cloud for all the applications involved. Based on the requirements of individual applications, the cloud infrastructure forwards a subset of the data only to the concerned applications. In a framework for connecting wireless sensor networks to the cloud was proposed for several community based applications. The work focused on developing a publish / subscribe system for efficient, distribution of data in the cloud, as well as on promoting cooperation between different cloud providers. In contrast, our focus is to develop a framework for unified sensing, and to provide application developers with a complete solution for robust and flexible deployment of people centric applications in dynamic environments.
As the rapid growth of the penetration rate of smart phones is expected, on-demand sensor information transmission and the reduction of the management cost are required. Consequently, the requirements for sensor data storage systems used by such services are as follows:

• Storing sensor data without compromising scalability.
• Tracking the data stored in mobile devices.
• On-demand sensor data transmission.

Sensor Data Repository Systems Ear-Phone or other existing services typically use the concentrated data storage method. In the concentrated method, mobile devices gather the necessary sensor information by service providers and upload these to concentrated storages. Using this method makes it easy for service providers to manage data flow. However, they end up spending more costs on storages due to the increase in the number of users participating in the sensing. To solve the problem, distributed sensor data sharing methods using peer-to-peer (P2P) networks have been proposed. P2P network nodes manage routing tables evenly to share the computation and communication cost. There are two types of P2P networks for implementing a sensor data storage system. One is constructing a P2P network among mobile devices and storing sensor data in the network. For example, LL-Net constructed a P2P network among mobile devices. LL-Net provided geographical range search capability. The other is constructing a P2P network between PCs owned by users and storing sensor data in the network. As an example, Mill provides a range search to retrieve sensor data taken within the range. The distributed method can share the storage cost by storing sensor data in the P2P network. On the other hand, the method LL-Net adopts needs to consider the characteristics of mobile devices such as network instability. When network connectivity is unstable, the management cost of the P2P network is undeniable. Furthermore, the method employed by Mill requires mobile devices to upload sensor data, therefore the on-demand sensor data transmission is compromised.

3. MPS DATA STORE

![Figure 2: An Overview of MPS Data Store](image)

We propose a sensor data repository system, MPS Data Store that will collect and store sensor information through users’ mobile devices. These mobile devices obtain information every minute, and attach the time and location of where and when the data is acquired. The mobile device stores all sensor data and transmits these only upon receiving queries from the service provider. MPS Cloud uses the IaaS cloud and enables geographical range and time range search functions. MPS Data Store enables on-demand sensor data transmission by separating the search function and storage function between the MPS Cloud and the mobile devices. As shown in Fig. 2, the mobile device travels along the movement trajectory of the user and senses data every minute. Immediately after sensing, the mobile device obtains the geographical location using GPS. Then, the mobile device converts the longitude and latitude information into an Area ID by using MPS Locator. An Area Notification is the act of notification of sensor data acquisition in the area. If the mobile device has not sent an Area Notification within the perimeter, it will send this notification to the MPS Cloud. Mobile devices’ IP address and Area IDs have to be included in an Area Notification. It is then transferred to an MPS Node, which manages the particular area. An MPS Node is a virtual machine dynamically created by the MPS Cloud and manages the Device Table. The Device Table includes information on which mobile device has the sensor data and in which area the information was acquired. The MPS Cloud and all of the mobile devices have an PS Locator, and all have the same scheme of ID allocation of areas and parameters. The MPS Cloud will know, therefore, which mobile device has the sensor data and in which area it was sensed using the MPS Locator.

4. IMPLEMENTATION

We implemented a prototype system that offers the basic functionalities of the proposed framework. The components residing at the cloud have been realized by using Amazon Web Services (AWS) and Google Fusion Tables. The Mobile Cloud Application (MCA) was built for Android devices running the 2.1 Platform and the AWS SDK for Android. Additional details are provided below.
5. Mobile Cloud Engine (MCE)

The Publish/Subscribe (PS) module was implemented by using Amazon’s Simple Notification System (SNS) and Simple Queue Service (SQS). SNS allows a cloud application to easily send notifications to the subscribers. SNS exploits the concept of topic to broadcast a notification from publishers to subscribers. In this context, the publisher is the application that sends messages, while the subscriber is the application (e.g., the one running on the mobile phone) that can receive the messages. A topic is an access point which defines specific subject or event type for publishing message sand allows clients to subscribe for notifications. In other words, the topic is the connection point between the sender and receivers. The notification can be in the form of SMS, email, SQS, and so on. In our framework, the MCE creates the topic and the mobile client application subscribes (unsubscribes) the mobile phone to (from) the topic. The framework uses the SQS as publisher and subscriber to a topic. SQS is an asynchronous mechanism by which the data or messages can be transferred between distributed components even without losing messages or requiring each component to be always available. For example, the mobile cloud application uploads the GPS data to the SQS and the MCE fetches the data from the SQS when needed at any time. SQS simplifies the interactions between the mobile cloud applications and the MCE. Furthermore, it supports read (and write) operations from (to) a large number of mobile cloud applications. Moreover, it provides security by authentication method. Using SQS in our framework enables mobile phones to send a large number of messages in parallel. In addition, SQS and SNS together enable the MCE and mobile client applications to send messages to a huge number of users in multiple formats quickly and easily. When the MCE needs to send a message to a group of mobile phones, it publishes the message to a topic. Once the messages have been published to a topic, all the subscribers of the SQS get the messages. Since, the mobile client application of each phone regularly polls the SQS, once the message is available in SQS, it can get the message. The context-awareness module is realized with reference to location-based information and is implemented through the Google Fusion Tables API. Google Fusion Tables provide mechanisms for publishing, retrieving, and rendering location-based data. It has been chosen for its features of scalability, accuracy, and availability, as well as for the built-in visualization based on Google Maps. Google Fusion tables are controlled by a Google account such that each account has its own fusion tables. In the proposed framework, the MCE was connected to a Google account in order to be able to use these features. Whenever the MCE has to return the result to the application community in the form of maps, it uses Google Fusion Tables. Specifically, the MCE writes the result into a fusion table, and based on those data, renders the map by using the Google Fusion API. Once the map has been generated, the MCE returns the obtained map and displays it through the widget. In addition, the data in the fusion table is continuously updated by MCE upon arrival of new data. As a result, the module is capable of rendering dynamic information to the map. In the current implementation, the decision module is limited to initialize the MCE, instantiate the PS modules, and handle the transactions between other modules.

6. EXPERIMENTAL ANALYSIS

In order to characterize the advantages of using our framework, we carried out two sets of experiments and studied the impact of the unified sensing approach in terms of energy consumption of the mobile device as well as scalability. In the first set of experiments, we assumed that the proposed framework is deployed in the cloud and is exploited by 4 applications. A thin mobile client installed on the mobile device collectively sampled and uploaded data to the UAM on behalf of the 4 applications. In the second set of experiments, we did not exploit our framework, but rather installed individual applications on the mobile device. We varied the number of applications from 1 to 4. They sampled and uploaded data individually on the mobile device. In all the experiments, data was captured and uploaded to the cloud every second. The evaluation was conducted by using a Google Nexus One smartphone. The duration of the experiment was set to 1,000 seconds. We measured the energy consumed by the mobile device for uploading messages in each experiment through the Power Tutor energy profiler application. The obtained results are shown in Figure 3. Each data point represents the average energy consumption over 10 iterations of each experiment, while the error bars represent the related confidence interval with a 95% confidence level. Figure 3 shows that the energy consumption of the mobile devices increase with the number of applications. Even though different applications require the same data, such as GPS coordinates or images, the device separately samples the data for each of them. Hence, as a result, it leads to same data being separately sent to each application. Thus, the mobile device experiences higher energy consumption with the increase in the number of applications. In contrast, the same data required by multiple applications are captured and sent to the UAM only once in our framework. The MCE collects and processes the data from each UAM and forwards it to the relevant application. Thus, this results in a lower energy consumption for the device, as it is not interacting with each application individually. Furthermore, we carried out additional experiments to evaluate the scalability and availability of the implemented prototype as a function of the number of users. To emulate users, we deployed 20 virtual machines sending an aggregate message rate ranging from 400 messages/sec to 4x10^5 messages/sec. We found that no messages were delayed at the message queues, thus we can conclude that the prototype is capable of handling a large number of users.
7. CONCLUSION
In this paper, we presented a sensor data repository system in Mobile Cloud and we presented an integrated framework to enable cloud-based and people-centric applications on mobile devices on Sensor Repository System. Our solution exploits the cloud computing paradigm to provide a new delivery model specifically targeted to mobile applications. Through a prototype implementation, we have shown the feasibility of the proposed framework by addressing a case study represented by an emergency response system. Experimental results demonstrate that the proposed approach results in significant energy saving at the mobile device. As a future work, we intend to add more features and functionalities to the proposed framework, with focus on the decision module.

References

AUTHOR
M.Chithik Raja received M.E. degrees in Computer Science and Engineering from R.V.S.C.E.T. affiliated by Anna university, Tamil Nadu, India in 2007. He spent around 8 years in india and working as lecturer in different colleges. After that he spent two years working in Ethiopian Institute of Technology (formerly it is called Mekelle University) Ethiopia. Book publishing where he was a publicist for Introduction to Management Information System, JAVA Programming for Beginners, Object Oriented Programming in C++ with Lampert Publication Germany. After moving to Sultanate of oman Working briefly as Lecturer and research fellow in Wireless Sensor Networks With Mobile Cloud Computing. He has published more than 10 International reputed Journal.