19th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD2015), Calabria, Italy, May 2015. © IEEE 2015

A Framework for Convergence of Cloud Services and Internet of Things

Dong Yuan, Jiong Jin, John Grundy and Yun Yang
School of Software and Electrical Engineering
Swinburne University of Technology
Melbourne, Australia
{dyuan, jiongjin, jgrundy, yyang}@swin.edu.au

***Abstract*—Today, Cloud Computing and the Internet of things are two "major forces" that drive the development of new Information Technology (IT) solutions. Many Internet of things (IoT) based large-scale applications rely on a cloud platform for data processing and storage. However, big data generated or collected by large-scale geo-distributed devices needs to be transferred to the cloud, often becoming a bottleneck for the system. In this paper, we propose a framework that integrates popular cloud services with a network of IoT devices. In the framework, novel methods have been designed for reliable and efficient data transportation. This framework provides a convergence of cloud services and devices that will ease the development of IoT based, cloud-enabled applications. We have implemented a prototype of the framework to demonstrate the convergence of popular cloud services and IoT technologies.**

Keywords*—*cloud computing; Internet of things; Service oriented architecture; Data transportation

I. INTRODUCTION

Cloud computing is widely considered to be the next generational step in Information Technology [3]. In recent years, cloud computing has emerged as the latest distributed computing paradigm that provides redundant, inexpensive and scalable resources on demand to meet system requirements. Since late 2007 when the concept of cloud computing was proposed, it has been utilised in many areas with considerable success [15]. Cloud computing adopts a pay-as-you-go model where users are charged according to the usage of cloud services such as computation, storage and network services like conventional utilities in everyday life (e.g. water, electricity, gas and telephony) [5]. Many IT resources could be delivered effectively and efficiently as a cloud service. By utilising cloud services, businesses can set up their required IT infrastructure easily, almost instantly and with minimal in-house IT expertise. Much research has been done with regard to resource allocation and cost optimisation for cloud based applications [16] [17] [14].

Meanwhile, the "Internet of Things" (IoT) [4] is becoming an increasingly popular research and practice area. IoT is a radical evolution of the current Internet into a network of interconnected objects. These not only harvest information from the environment (sensing) and interacts with the physical world (actuation/command/control), but also uses existing Internet standards to provide services for information transfer, analytics, applications and communications [2]. As the capabilities of smart devices advance (in terms of processing power, network connectivity and sensors), people increasingly use them for building all kinds of applications in different sectors. The proliferation of IoT with smart devices leverages the efficiency for applications where users can access and share information location independently. However, these devices are often geo-distributed and generate or collect large volumes of data that need corresponding resources for processing and storage. Although there have been many advances, such devices will always face resource limitations such as restrictions on weight, size, battery life, and heat dissipation. These impose limitations on computational and storage resources and make devices more resource constrained than their non-mobile and larger counterparts [6].

In light of this, many IoT applications often employ the cloud computing paradigm to manage their data processing and storage needs [7]. This is often referred to as Cloud-centric IoT, where the Cloud integrates all facets of ubiquitous computing by providing scalable storage and computational resources to build new businesses. Moreover, the core objective of the Cloud to efficiently model cost based on supply and demand offers a unique opportunity to create an efficient IoT business model [11]. In this model, IoT application providers can join the network and offer their data using a storage Cloud; analytic tool developers can provide their software tools; and computational intelligence experts can provide their data mining and machine learning tools useful in converting information to knowledge. Cloud computing is able to offer these services as Infrastructures, Platforms or Software. Specifically, the data generated, tools used and algorithms developed are all hidden in the background, with focus given to various application domains of IoT. However, several challenges exist in such Cloud-based IoT applications. Specifically, 1) big data generated or collected by devices needs to be transferred to the cloud; and 2) we need to integrate cloud services to the IoT network for application developers to use. Current solutions and approaches do not adequately address these challenges.

In this paper, we propose a novel framework that integrates cloud services with Internet of things networks. Our framework can optimise the data transportation from geo-distributed devices and the cloud. The framework also provides unified APIs (Application Programming Interface) and SDKs

(Software Development Kit) with the convergence of devices and the cloud to developers for their ease of developing IoT based applications. We have implemented a prototype of this framework, which demonstrates the steps of the IoT based application's execution process with optimised data transfer to the cloud.

The remainder of this paper is organised as follows. Section II gives a motivating scenario for such a cloud-enabled IoT application and analyses key research problems. Section III presents the details of our framework for the convergence of cloud services and IoT. Section IV describes a prototype implementation of our framework and our evaluation of this prototype. Section V summarises our conclusions and highlights key directions for future work.

II. Motivation and Problem Analysis

We introduce the concept of the Smart City as a motivating scenario, where IoT based applications are widely utilised. Based on this scenario, we present key research problems and questions in realising this application.

A. Motivating Scenario – the Smart City

A "Smart City" is the one that uses information and communications technologies to make the city services and monitoring more aware, interactive and efficient. This is driven and enabled technologically by the emergence of the Internet of Things (IoT) [4]. A smart city utilises information and communications technologies (ICT) in a way that addresses quality of life challenges by tackling urban living challenges encompassed by more efficient utilization of limited resources (space, mobility, energy, etc.). World leading municipalities, in terms of services and quality of life, have provided efficient services to their citizens by forward thinking and use of technology in monitoring various environmental parameters.

Most of these systems consist of: sensor, data storage device, and computer at a base station where experts analyse the data. Our close interaction with the City of Melbourne, has allowed us to unearth the vast ICT potential in making the entire system more efficient. The applications within the urban environment that can benefit from a smart city IoT capability can be grouped according to impact areas. This includes the effect on: citizens (health and wellbeing); transport (mobility, productivity, pollution); and services (critical community services). Several projects are already underway within the City of Melbourne that utilize sensor technologies to collect application specific data. These include: public parking monitoring; microclimate monitoring; access and mobility (pedestrian, cyclists, cars and freight vehicles), and real-time public transport data. A number of specific application domains have also been identified that could utilize smart city IoT infrastructure to service operations in: Health Services (hospitals and personal care); Environmental (better noise, air and water quality); Strategic Planning (enhanced personal and business mobility); Sustainability and Utilities (better energy usage); Tourism (improved visitor services, tourist activity); Transport (personal and logistics); Business and International (city usage, access); and City Safety.

From the technological perspective, the evolution of social networking in the past decade clearly shows the usability of ICT at an individual level. Large-scale implementations at a system level have also made some progress in recent years. A fully integrated system of systems containing sensing, storage, analytics, and interpretation is now required. The integrated system must have core capabilities of plug-and-play sensing, secure data aggregation, Quality of Service, and re-configurability. With an urban sensing system of systems in place, the ability to evaluate the impact of the preceding actions is readily available as the sensing cycle repeats.

A unifying information management platform delivers a capability across application domains critical to the city. Whilst large volumes of data collection and interpretation are already performing at different levels within city councils using manual and semi-automated methods, it is mostly in isolation. As with any large organization, it is inevitable that large portions of these data remain disjoint. An urban information framework enabled by IoT provides a means for consolidating these tasks and sharing of data between various service providers in the city.

B. Problem Analysis

The realisation of smart city needs to be based on IoT technologies, where a very large number of heterogeneous, geo-distributed devices is used for data collection and control. Due to the limited capacity of most such devices in terms of computation, storage and energy, the generated or collected data needs to be effectively and efficiently transferred to a cloud application for processing and storage. However, the devices (which in many cases are small sensors) often only have a local area network (LAN) connected with other devices in the field and do not have Internet access. Hence, data in these devices needs to be collected periodically and uploaded to the cloud. For applications that need real time data processing, the generated application data has to be transferred to the cloud as fast as possible. Some IoT devices even need cloud computational services to process large data and return results to the device e.g. to allow them to control local hardware systems. How to guarantee the reliable and efficient data transfer from large number of geo-distribute devices to cloud services and back is the **first** research problem of this paper.

Furthermore, smart city incorporates applications and devices from many domains. For developers, they not only need to understand the requirements of the application, but also have to cope with the diverse hardware and APIs of both cloud services and the IoT network. This is a big challenge for developers as they need to learn the technical specifications of different cloud services and IoT technologies. Hence our **second** research problem is to simplify the development process of such cloud-enabled IoT applications for developers.

In the next section, we present the design of a new framework supporting the convergence of cloud services and IoT, that address our above two key research problems.

III. Framework

We first briefly overview the architecture of IoT based applications that need cloud services for data processing and storage, and point out the main focus of our framework. We then describe our framework as a Service Oriented Architecture (SOA) and introduce how our framework can support developers. Last, we present the design of algorithms that guarantee the reliable and efficient data transfer from devices to the cloud.

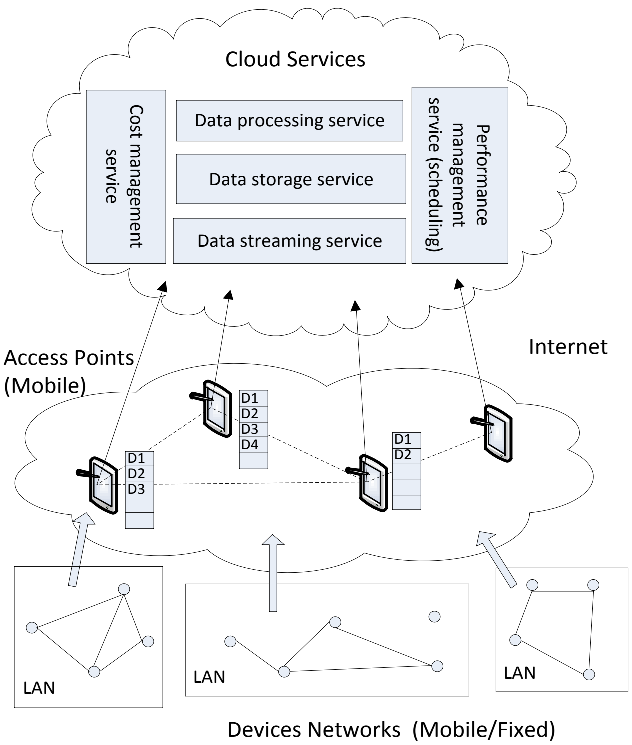

Figure 1. IoT Application and Cloud Services

A. Overview of Cloud Services and IoT Application

1) Cloud Services

In recent years, cloud computing has emerged as the latest distributed computing paradigm that provides redundant, inexpensive and scalable resources on demand to meet challenging and dynamic system requirements [9]. As IaaS (Infrastructure as a Service) is a very popular way to deliver computing resources in the cloud [1], the heterogeneity of computing systems of one service provider can be well shielded by virtualisation technology. Hence, users can deploy their applications in unified resources without any infrastructure investment, where excessive processing power and storage can be obtained from commercial cloud service providers [12]. As cloud computing systems are usually based on the Internet, users can upload their data and launch their applications in the cloud from anywhere in the world via the Internet.

As shown in Figure 1, to support processing and storing data from IoT based applications, different types of cloud services are utilised. A data streaming service is provdied for hosting large data stream from different devices. It can cache and initially filter the raw data collected from devices. The data storage service is for storing all types of application data (either initial data or analysis results) for different purposes (either temporal or permanent storage). A data processing service is used for all types of computation tasks, e.g., normal CPU instance for hosting the application, Map-Reduce cluster for data analytics, etc.

2) Device Networks of IoT

There are two main design approaches for the network architecture: (1) an evolutionary approach; and (2) a clean-slate approach [13]. The evolutionary approach makes incremental changes to the current network architecture to reuse as many components as possible from existing networking solutions. From this perspective, an IoT could be viewed as an extended architecture evolved from the Internet. On the other hand, the clean-slate approach advocates a re-design of network without being constrained by the current structure. It means, in order to cope with next-generation network challenges, new architecture and protocols will be developed according to disruptive design principles. Indeed, an ongoing debate about these two approaches has emerged in the networking research community over the past several years. Ultimately, individual researchers have their own styles, often a unique blend between them as the applications dictate [13].

As shown in Figure 1, the device networks of IoT application are large-scale and geo-distributed LANs. In each LAN, devices are interconnected with a variety of low power communication protocols (e.g., Blutooth 4.0, Zigbee, NFC, etc.). Due to energy limit and cost issues, these devices normally do not have direct Internet access, but they keep generating application data and store it temporarily locally. In the IoT application, these data need to be collected and transferred to the cloud periodically.

3) Mobile Access Points for Data Collection

In order to bridge the gap between cloud services and device networks of IoT, we introduce another layer of mobile access points for data collection and uploading. The mobile access point is a smart device that has Internet access (normally via a telecommunication network 3G/4G) and travels among LANs of devices.

As shown in Figure 1, whenever the mobile access point enters a LAN of the application, it collects all generated application data of devices in this LAN. The data are kept in a queue in the mobile access point for uploading to the cloud. As the mobile access points are travelling around, whenever they are close, they form a network (ad hoc/P2P) and connect with each other. Normally, the bandwidth between each access point and the cloud is limited and the data for uploading are very huge. Hence, whenever some mobile access point join a same network, they will swap their uploading queue to balance their workload.

In Section III.C, we will introduce time stamp-based algorithms for reliable data uploading which guarantees no duplicated data will be uploaded, and a optimisation model for the total uploading time of all application data.

B. Architecture of the Framework

The layers of our framework are shown in Figure 2, which has a service oriented architecture (SOA) with elasticity and scalability.

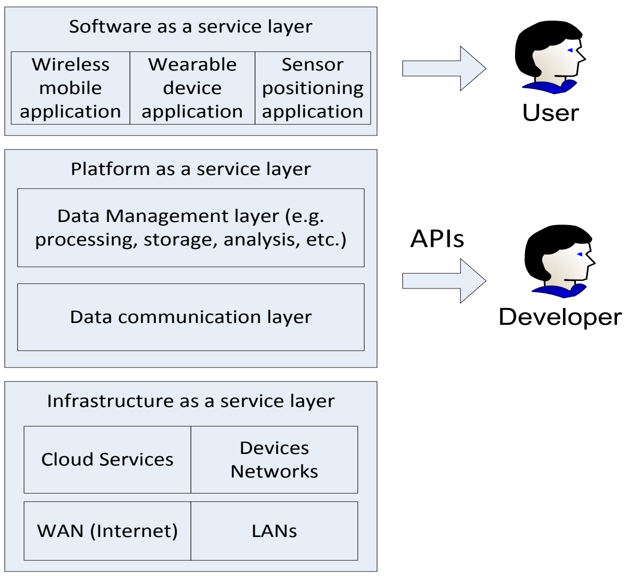

Figure 2. Service Layers of the Framework

1) System Layers for SOA

SOA has been used for building devices-based networks [8]. With SOA, our framework can incorporate cloud services in different forms, i.e., IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service). Cloud services can dynamically scale up and down according to the workload and the number of devices connected to the platform. We mainly adapt two different technologies.

(1) Service selection and combination technology. The service selection problem has been an active research topic for some years [10]. In our framework, we adapt existing technologies to access popular cloud services from different providers. According to different functions and levels of cloud services, they can be combined with different devices in different applications. As shown in Figure 2, in the IaaS layer, different cloud services and devices networks are integrated together.

(2) QoS management technology. As cloud services are combined with devices, QoS management is a big challenge for the platforms, especially for communication between devices and the cloud. We develop a QoS management mechanism from monitoring and prediction to violation detection and adjustment. We also develop a lifetime QoS assurance mechanism for the convergence of cloud services and smart devices.

2) Unified APIs and SDKs for Developers

One of the purposes of our framework is to ease development of IoT and cloud services-based applications. With unified APIs and SDKs, developers can directly access cloud services and smart devices without knowing their different hardware and networking specifications. In other words, they can develop applications incorporating different cloud services and smart devices with a unified suite of APIs.

In order to abstract unified APIs and SDKs that shield the differences of cloud services from different providers and the difference of smart devices, first, we investigated SDKs of existing cloud services and the popular framework for mobile computing and Internet of things. Existing popular public cloud service providers include Amazon, Microsoft, Google, etc. Each provider has a bunch of different cloud services in different levels (IaaS, PaaS and SaaS). We analysed their differences and classified them according to their functions. A representative of the Internet of things framework is AllJoyn project [1], which provides a unified language of different communication protocols for different devices. With the AllJoyn framework, smart devices can form an ad hoc peer to peer local area network.

Then, we do a second phase of API and SDK development based on those SDKs and projects, and abstract the unified APIs for users according to their requirements. For users from different domains, our platform has a different set of SDKs for them to develop their applications.

C. Connection between Devices and the Cloud

In this sub-section, we describe the algorithms for reliable and efficient data transfer from large number of geo-distribute devices to cloud services.

In order to present the algorithm, we introduce some denotations as follows. We use dc_i to denote the geo-distributed devices, AP_i to denote the mobile access points, B_i to denote the uploading bandwidth of AP_i to the cloud, $d_i^{\mathrm{dc}_j}$ to denote a piece of data generated by device dc_i, furthermore $d_i^{\mathrm{dc}_j}.\mathrm{size}$ is the size of $d_i^{\mathrm{dc}_j}$ and $d_i^{\mathrm{dc}_j}.t$ is the uploading deadline of $d_i^{\mathrm{dc}_j}$.

1) Time Stamp based Algorithm for Reliable Data Transfer

In our scenario, a large number of geo-distributed devices in different LANs generate large volumes of data, but the bandwidth between the mobile access points and the cloud is limited. Due to this fact, every mobile access point normally has a long queue of data waiting for upload. Furthermore, as the mobile access points need to travel around the field to collect data, we cannot guarantee that an access point finishes all the data uploading for the LAN before leaving it. Hence, we develop a time stamp based algorithm to guarantee the reliable data transfer. The algorithm works in the following steps.

(1) For a device, denoted as dc_i, it places a time stamp for every piece of generated data.

[1] https://www.alljoyn.org/

(2) In the cloud, it records the most recent uploaded data for dc_i.

(3) Whenever dc_i detects a mobile access point, denoted as AP_j, enters its LAN, checks for the already updated data in the cloud through AP_j. Then it deletes all the updated data and sends all non-updated data with the time stamp to the queue in AP_j.

(4) For a mobile access point, denoted as AP_j, it keeps uploading data in the queue to the cloud. Specifically, whenever it starts uploading data from device dc_i, it first checks the most recent uploaded data from dc_i in the cloud. If some data from dc_i in the queue have already been uploaded, then deletes the uploaded data and starts uploading from the most recent non-updated data.

Based on the above algorithm, we can guarantee that all the generated data with no duplications will be uploaded to the cloud.

2) Bandwidth Optimisation Model for Efficient Data Transfer

Mobile access points will have different bandwidth to the cloud (e.g., 3G/4G), hence the length of uploading queues also changes. However, every piece of data generated by devices has a deadline for its upload. Hence, for a mobile access point, we need to guarantee that all data can be uploaded before deadlines. Formally, for a piece of data $d_i^{\mathrm{dc}_j}$, we need to guarantee that

$$\mathrm{t} + t_{current} < d_i^{\mathrm{dc}_j}.t$$

$$\Rightarrow \frac{\sum_{h=1}^{i}(d_h^{dc_j}.size)}{B} + t_{current} < d_i^{dc_j}.t,$$

where *t* is the waiting time for $d_i^{\mathrm{dc}_j}$ in the queue.

If a mobile access point has data in its uploading queue that violates their uploading deadline, we have to do balance scheduling whenever it joins the network.

Furthermore, for a mobile access point, denoted as AP_j, the total time for its uploading queue is $\mathrm{T}_j = \frac{\sum_i(d_i^{\mathrm{dc}_k}.size)}{B_j}$. Whenever some mobile access points are in a LAN, we can do the balance scheduling to reduce the total uploading time for the heavily loaded mobile access point.

IV. Implementation and Evaluation

We have implemented a prototype of our framework based on popular cloud services and devices network of IoT. In this section, we first introduce the popular technologies adapted in our implementation, and then present a sample case in the smart city scenario to evaluate our framework.

A. Amazon Cloud Sevices

As Amazon is a well-known and widely recognised cloud service provider, we implemented our framework based on Amazon cloud services.

For the data streaming service, we use Amazon Kinesis[2] to get data from all the mobile access points. Amazon Kinesis is a fully managed service for real-time processing of streaming data at massive scale. Amazon Kinesis can continuously capture and store terabytes of data per hour from many sources such as website clickstreams, financial transactions, social media feeds, IT logs, and location-tracking events.

For the data storage service, we use Amazon S3[3] to storage all types of application data. Amazon Simple Storage Service (Amazon S3), provides developers and IT teams with secure, durable, highly-scalable object storage. Amazon S3 is easy to use, with a simple web services interface to store and retrieve any amount of data from anywhere on the web.

For the data processing service, we use Amazon EMR[4] for data analysis and EC2 for hosting the application. Amazon Elastic MapReduce (Amazon EMR) is a web service that makes it easy to quickly and cost-effectively process vast amounts of data. Amazon EMR uses Hadoop, an open source framework, to distribute your data and processing across a resizable cluster of Amazon EC2 instances. It can also run other distributed frameworks such as Spark and Presto. Amazon EMR is used in a variety of applications, including log analysis, web indexing, data warehousing, machine learning, financial analysis, scientific simulation, and bioinformatics.

B. AllJoyn Framework for Devices Network

For the device network of IoT, we adapted the popular AllJoyn framework, a common language for the "Internet of Everything". AllJoyn is an open source project that lets compatible smart devices recognize each other and share resources and information across manufacturer, networks, and operating systems. Products, applications and services created with AllJoyn can communicate over various transport layers, such as Wi-Fi, power line or Bluetooth, regardless of manufacturer or operating system and without the need for Internet access. The software runs on popular platforms such as Linux and Linux-based Android, iOS, and Windows, including embedded variants.

In our prototype, we used AllJoyn to create an ad hoc/p2p LAN for devices connected with Bluetooth protocol. For data collection and uploading, we use Android smart phones as the mobile access points and also create a Bluetooth LAN with AllJoyn for all the smart phones. As access point, every smart phone has Internet access and is connected to Amazon Kinesis service for data streaming.

C. A Case Study for Evaluation

Smart traffic monitoring system is a typical IoT based application in smart city, where users can get real time information and statistics by analysing the real time data collected by devices. A simple example is that the user wants

[2] http://aws.amazon.com/kinesis/
[3] http://aws.amazon.com/s3/
[4] http://aws.amazon.com/elasticmapreduce/

to know how many accidents happened in the last one hour in the whole city. Figure 3 demonstrates the execution process of this requirement in our framework.

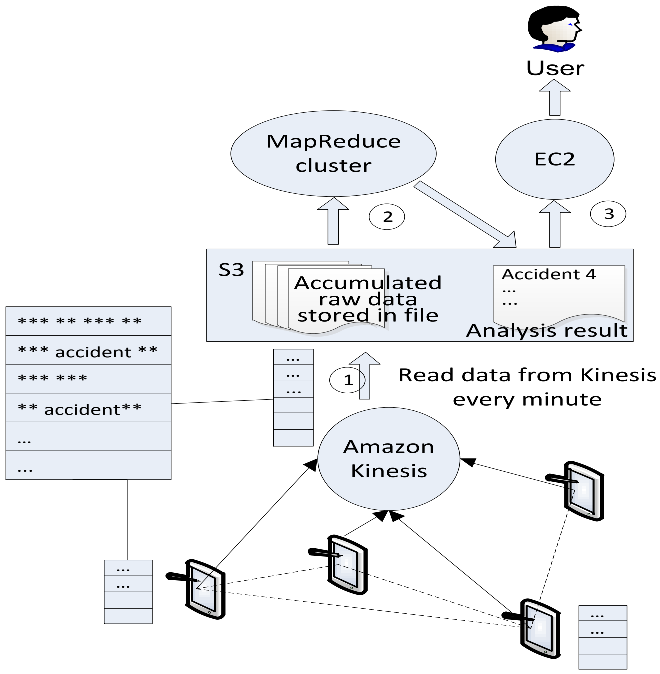

Figure 3 Execution process of our framework

The first step is data collection. Accident information is collected by geo-distributed devices (sensors or cameras), transferred to mobile access points, and then uploaded to Amazon Kinesis. The application reads data from Kinesis every minute and stores it as a file in Amazon S3. The second step is data analysis. According to the requirement, data files recorded in the last hour are sent to Amazon MapReduce cluster for analysis, i.e. counting the number of accidents recorded in the files. Then the results will be sent back to S3 for storage. The third step is result demonstration. Users can view the result through the application.

V. CONCLUSION AND FUTURE WORK

In this paper, we investigated the issue of the gap between cloud servicse and devices in IoT applications. We propose a new framework with a prototype implementation that integrates popular cloud services (i.e. Amazon cloud services) with the network of IoT devices (i.e. AllJoyn network. Novel methods have been designed for reliable and efficient data transportation. The framework with the convergence of cloud services and devices can provide developers an SDK for easier development of I cloud-enabled, IoT-based applications. Evaluation of our prototype demonstrated the convergence of popular cloud services and IoT technologies is feasible. In the future, large-scale experiments need to be conducted to evaluate the scalability of our framework and its generality.

ACKNOWLEDGMENT

The research work reported here is partly supported by Australian Research Council under LP130100324. We are grateful for discussions on the ideas in this paper with Dr. Quoc Bao Vo from Swinburne University of Technology.

REFERENCES

[1] *Amazon Cloud Services*. Available: http://aws.amazon.com/
[2] "Smart Networked Objects and Internet of Things," *White Paper, Association Institutes Carnot, Greece,* 2011.
[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski*, et al.*, "A View of Cloud Computing," *Communication of the ACM,* vol. 53, pp. 50-58, 2010.
[4] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks,* vol. 54, pp. 2787-2805, 2010.
[5] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Future Generation Computer Systems,* vol. 25, pp. 599-616, 2009.
[6] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proceedings of the sixth conference on Computer systems*, 2011, pp. 301-314.
[7] G. Fortino, A. Guerrieri, W. Russo, and C. Savaglio, "Integration of Agent-Based and Cloud Computing for the Smart Objects-Oriented IoT," in *IEEE 18th International Conference on Computer Supported Cooperative Work in Design*, 2014, pp. 493-498.
[8] G. Fortino, D. Parisi, V. Pirrone, and G. Di Fatta, "BodyCloud: A SaaS approach for community Body Sensor Networks," *Future Generation Computer Systems,* vol. 35, pp. 62-79, 2014.
[9] I. Foster, Z. Yong, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," in *Grid Computing Environments Workshop*, Austin, Texas, USA, 2008, pp. 1-10.
[10] G. Fox, S. Pallickara, and S. Parastatidis, "Toward Flexible Messaging for SOAP-Based Services," in *Proceedings of the ACM/IEEE Supercomputing Conference,* 2004, pp. 1-11.
[11] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An Information Framework for Creating a Smart City through Internet of Things," *IEEE Internet of Things Journal,* vol. 1, pp. 112-121, 2014.
[12] X. Liu, Z. Ni, Z. Wu, D. Yuan, J. Chen, and Y. Yang, "A Novel General Framework for Automatic and Cost-Effective Handling of Recoverable Temporal Violations in Scientific Workflow Systems," *Journal of Systems and Software,* vol. 84, pp. 492-509, 2011.
[13] J. Rexford and C. Dovrolis, "Future Internet Architecture: Clean-Slate Versus Evolutionary Research," *Communications of ACM,* vol. 53, pp. 36-40, 2010.
[14] D. Yuan, Y. Yang, X. Liu, and J. Chen, "A Cost-Effective Strategy for Intermediate Data Storage in Scientific Cloud Workflows," in *24th IEEE International Parallel & Distributed Processing Symposium*, Atlanta, Georgia, USA, 2010, pp. 1-12.
[15] D. Yuan, Y. Yang, X. Liu, and J. Chen, "A Data Placement Strategy in Scientific Cloud Workflows," *Future Generation Computer Systems,* vol. 26, pp. 1200-1214, 2010.
[16] D. Yuan, Y. Yang, X. Liu, and J. Chen, "On-demand Minimum Cost Benchmarking for Intermediate Datasets Storage in Scientific Cloud Workflow Systems," *Journal of Parallel and Distributed Computing,* vol. 71, pp. 316-332, 2011.
[17] D. Yuan, Y. Yang, X. Liu, W. Li, L. Cui, M. Xu*, et al.*, "A Highly Practical Approach towards Achieving Minimum Datasets Storage Cost in the Cloud," *IEEE Transactions on Parallel and Distributed Systems,* vol. 24, pp. 1234-1244, 2012.