A Cloud You can Wear: Towards a Mobile and Wearable Personal Cloud

Ragib Hasan and Rasib Khan
SECRETLab, Dept. of CIS, University of Alabama at Birmingham, AL, USA
Email: {ragib, rasib}@cis.uab.edu

***Abstract*—Mobile and wearable devices provide the expected user experience and the ability to run complex applications using cloud based services. This makes the design of such wearable devices complex, expensive, and with major data privacy concerns. In this paper, we present the concept of a wearable cloud – a complete yet compact and lightweight cloud which can be embedded into the clothing of a user. The wearable cloud makes the design of mobile and wearable devices simple, inexpensive, and lightweight, tapping into the resources of the wearable cloud. We introduce five wearable cloud service delivery models including a prototype implementation of the wearable cloud and a cheap touchscreen terminal device. The paper presents experimental results on the usability of the wearable cloud based on energy consumption and application performance.**

***Keywords*-Wearable Cloud, Cloud Jacket, Mobile Device, Wearable Device, Mobile Cloud, ChipCloud, Personal Cloud**

I. Introduction

Personal wearable and mobile devices have changed the world of computing in recent years [1, 2]. However, applications are becoming more computationally intensive, mandating a continuous upgrading of hardware configurations for such devices, resulting in increased prices. As a result, mobile applications are powered by the cloud to overcome the resource limitations. Unfortunately, a lot of computations are still required to be performed on the devices. Additionally, public cloud providers mandate the transfer, storage, and routing of personal data in remote locations via public networks.

At the same time, mini computers are becoming popular for various standalone Internet-of-Things applications and are featured with tiny processors and on-board memory. Such devices include the Raspberry Pi[1a], Banana Pi[1b], BeagleBoard[1c], and the Intel Galileo[1d]. They require low operating power, have a small form-factor, and are available for cheap prices [3].

In this paper, we present the concept of a wearable cloud, as illustrated in Figure 1. The proposed concept leverages the use of mini computers mounted on a jacket, a briefcase, a back-pack, etc., to overcome the limitations of mobile and wearable devices using public clouds. The wearable cloud is similar to a body area network of mini computing nodes, with minimal power requirements, and elastic resource provisioning on the private wearable cloud [4–6]. The requirement based provisioning of resources allows conservative energy usage and ensures proximal placement of personal data. As a result, the mobile and wearable devices can be turned into dumb user-interactive terminals.

The wearable cloud is not just a wearable computer or a powerful laptop. The MIT Smart Vest [7] can be considered as a predecessor of the wearable cloud. The Smart Vest focuses on the fabrication of the computing chips and devices. However, the proposed wearable cloud focuses on elastic resources and cheap terminal devices as service end-points.

[1](a) www.raspberrypi.org, (b) www.bananapi.org, (c) beagleboard.org, (d) www.intel.com/content/www/us/en/embedded/products/galileo

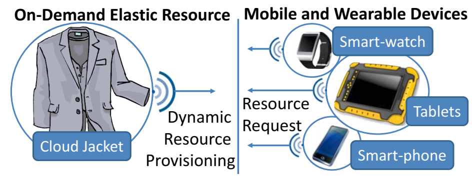


Fig. 1: Overview of the wearable cloud model.

The mini computers can be dynamically allocated to and shared by the terminal devices in the same manner as a cloud, maximizing resource utilization and allowing for bursts of computation demand via resource allocation and elasticity. The unused mini nodes can be powered down during low load to reduce the power utilization. The proposed framework is a fully functional wearable cloud that the user will carry around and use it like a real cloud via various personal mobile and wearable devices. The wearable cloud offers privately owned, proximal, and guaranteed resources for the user's devices, and thus, is different from traditional clouds or cloud-lets [8].

Organization: The motivation and background for the various concepts are presented in Section II. The system architecture and service delivery models are described in Section III and Section IV respectively. We discuss our prototype implementation of the cloud jacket and the experimental results in Section V. Section VI presents the related work, and we conclude with a discussion in Section VII.

II. Motivation

In this section, we present the challenges in mobile/wearable computing and the prospects of utilizing a wearable cloud.

A. Challenges in Mobile and Wearable Computing

Low computing capability: Wearable devices have processors that are almost 10 times or more slower than desktop/laptop processors, limiting the type of applications executable on such devices. For example, the LG Google Watch processor scored only 1198 at best using Geekbench 3 benchmark, while an iMac processor scored 14772 on the same benchmark [9–11].

Internet connectivity for offloading: The low computational capability of mobile devices are overcome by creating a hybrid application model where the bulk of the computation is offloaded to a cloud server. However, this necessitates constant communication over the Internet (via 4G/LTE or Wi-Fi).

Public clouds/cloud-lets and networks: Mobile and wearable clients are required to upload all personal data to physically distant public clouds or localized cloud-lets [8], without the knowledge of the physical location of the data. Moreover, all data to/from the personal devices are sent/received via public networks, introducing network latency and security concerns.

Power drain: The power drain on wearable device batteries is a major concern, hampering the user experience by requiring frequent recharging of the devices. For example, the Google Glass lasts between 2 to 5 hours with light usage, and only 30 to 60 minutes for video applications [12].

0730-3157/16 $31.00 © 2016 IEEE
DOI 10.1109/COMPSAC.2016.25

Complex design and cost: Powerful processors and user expectations for high performance applications have caused the design of wearable and mobile devices to be complex and expensive. A consumer wearing a smart watch and a smart glass, carrying a smart phone, and wearing wearable healthcare devices would have to spend between $2,000 and $3,000 for purchasing such devices [13, 14].

B. Solving the Challenges using a Wearable Cloud

The issues described above make mobile and wearable computing challenging to achieve in the real world. However, we can solve all these problems by bringing a cloud-like computational resource at close proximity to such devices. Since we are focusing on mobile and wearable devices, the computational resource would have to be mobile as well and must follow the user (and the devices). This motivates our research on a wearable cloud. We argue that a wearable cloud would solve the aforementioned problems as follows:

1) User's mobile and wearable devices can run complex/heavyweight applications in the wearable cloud.
2) Lower-cost (in terms of latency and energy) computation offloading to the wearable cloud from mobile devices will lead to longer battery lives.
3) All of the user's devices can share the same cloud, making communication, data and sensor-sharing, and cooperative applications efficient and easy to implement.
4) Mobile and wearable devices will no longer need to have complex and powerful processors. The wearable cloud will provide the experience of a smart device with simple, lightweight, and low-cost 'dumb' terminal devices, leading to widespread adoption by the general population.
5) Retain most, if not all, personal data in the user's proximity within the privately owned wearable cloud. This will reduce the network latency and remove the security concerns.

III. System Architecture

In this section, we present the model of the wearable cloud and may extend to anything and everything which can be carried around by a user as a personal carry-on item. It may be in the form of a jacket, a small brief-case, a back-pack, etc.

A. System Model

The wearable cloud is powered by a number of small computing nodes (e.g. Raspberry Pi) interconnected to each other, similar to body area networks [4–6]. The nodes can be powered using a portable/rechargeable power-bank. The mobile and wearable devices are turned into cheap terminals with minimal processing power. A user utilizes the mobile and wearable (terminal) devices to request services via user-intuitive display and interactions. The computational task is sent to the wearable private cloud. The nodes are engaged based on an elastic resource provisioning model, and computes the task collectively. Upon completion, the displayable result is sent back to the terminal device.

The 'cloud' analogy is introduced with the fact that only the required number of compute nodes will be engaged for the given task and only consuming the required amount of resources. The mobile terminals can use the wearable cloud to execute resource hungry applications, such as a data analytic software, and view the results on the mobile terminals. This

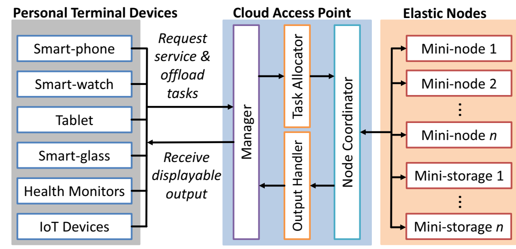


Fig. 2: Wearable cloud operational architecture.

is different from mobile computing or laptops, which require wholistic exhaustion of native resources (battery, memory, storage) for executing on-device applications.

B. Operational Architecture

The operational architecture for the wearable cloud is inspired by mobile and dynamic clouds [15–17]. The operational details for each of the components, as illustrated in Figure 2, are described as follows.

Personal Terminal Devices (PTD): The PTDs are basic interactive devices capable of wireless communications (e.g. WiFi, Bluetooth, ZigBee). Such devices include, but are not limited to, smart-phones, smart-watches, tablet computers, smart-glasses, health monitors, and other IoT devices. We posit that the hardware specifications on the PTDs can be lowered, to create rather 'dumb', cheap, and resource constraint devices, in terms of computational power, battery, memory, low-power short-range radio, and storage. The PTDs can cooperate via the wearable cloud for composite services.

Cloud Access Point (CAP): The CAP may be allocated as one of mini nodes dedicated to serve the wearable cloud. The CAP may be enough to serve the requests from the PTDs without further elastic resources. The CAP acts as the gateway for the PTDs to communicate with the wearable cloud for the user's service requests and offloaded tasks via wireless interfaces (e.g. WiFi, Bluetooth, ZigBee). The service requests and responses require organized management and elastic scheduling and coordination of mini nodes. The CAP consists of the following procedural components to facilitate the wearable cloud framework.

- **Manager (MGR):** The MGR receives the requests from the PTDs, keeps track of the progress of the request, and maintains the sessions for multiple PTDs and incoming requests. The output handler delivers the results to the MGR, and subsequently, the MGR delivers the formatted output results to the particular PTD(s).
- **Task Allocator (TAC):** The TAC is invoked by the MGR to estimate the required resources for the given service request. The TAC is responsible to parallelize the task (as much as possible) and divide the requested service task into chunks of parallel processes and input data according to the estimated required resources. The segmented tasks and input data are then passed on to the node controller for subsequence processing.
- **Output Handler (OHL):** The OHL obtains the result(s) from the node controller once a task is completed. The OHL generates the aggregated result and formats the output for the given context and device. Upon completion, the output is then handed over to the MGR.

- **Node Controller (NCR):** The NCR is responsible for handling the elastic mini nodes within the wearable cloud framework. The NCR manages the elastic resource allocation and provisioning. The segmented tasks and corresponding input data are obtained from the TAC. The NCR then allocates the elastic mini compute and storage nodes according to the requirements placed by the TAC. The subtasks are assigned to the allocated mini nodes and the subresults are collected upon completion.

Elastic Nodes: The elastic resources are mini nodes, in the form of computational and storage resources, using chip-based computers and flash-based drives. The mini compute nodes can be utilized to perform processing tasks and provide the private storage facility for the wearable cloud. The nodes communicate with each other via a common gateway with the CAP over a private wired or wireless network.

IV. Service Delivery Model

In this section, we propose and illustrate the use of the above system architecture for various applications and the corresponding service delivery models.

A. Thin Mobile and Wearable Clients

The wearable cloud leverages the concept of thin PTDs, i.e., user devices with minimal computation power and storage, low-power short-range radios, and are primarily featured as user-interfacing devices. Figure 3(a) illustrates a simplistic service delivery model for thin PTDs using the wearable cloud.

In the first scenario (Figure 3(a.1)), a user interacts via a thin PTD and creates $UsrIntr$ data. The service request, $ServReq$, is forwarded to the CAP. The CAP allocates the mini nodes using $AllocReq$ requests and $AllocConf$ confirmations for the given task. The sub-tasks are aggregated using the $P_{Aggr}(data_{alloc})$ process, and the service response, $ServResp$, is sent from the CAP to the thin PTD. At this point, the PTD merely formats the data and generates the graphical interface for the user using process $P_{GUI}(data_{ServResp})$. In case of notification-based services (Figure 3(a.2)), the service monitor $P_{ServMon}(Ev)$ in the CAP is triggered for any probable notification event. The CAP executes the task using the elastic mini nodes via $AllocReq$ requests and $AllocConf$ allocation confirmations. The $P_{Aggr}(data_{alloc})$ process aggregates the results, and sends the $ServNotf$ notification to the thin PTD(s). The PTD only formats the displayable data using the $P_{GUI}(data_{notf})$ process and notifies the user.

Application: *Alice owns a cheap PTD with a light-weight processor and low-power short-range radio, but can easily access and use advanced image recognition and voice-based search applications, harnessing the resources from the wearable cloud. She does not need to upgrade her phone every year to be able to enjoy the new mobile games which she likes to play on her phone.*

B. Elastic Mobile Infrastructure-as-a-Service

A simplified model for localized and mobile outsourcing of Infrastructure-as-a-service (IaaS) is illustrated in Figure 3(b). The wearable cloud can allow the PTDs to offload computational tasks using $RsrcReq$ requests (Figure 3(b.1)). The tasks are executed using the elastically allocated mini nodes, aggregated using process $P_{Aggr}(data_{alloc})$, and the results are

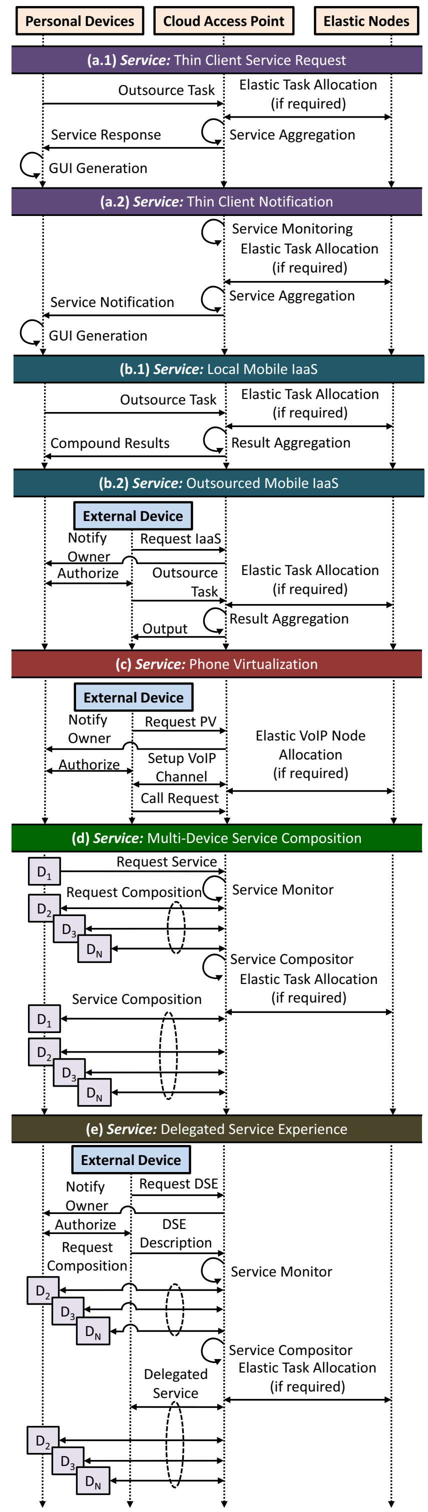


Fig. 3: Service delivery model for wearable cloud.

sent to the PTD via the $RsrcRes$ message. The wearable cloud owner may also allow outsourced external PTDs to offload computational tasks (Figure 3(b.2)). The external PTD makes a $RemReq$ request to a nearby wearable mobile cloud. The CAP notifies the owner on the PTD using $NotfRemReq$ to grant authorization to the external PTD. Once authorized, the co-located external PTD can offload the outsourced task to the CAP using $RemRsrcReq$. The wearable cloud elastically allocates the mini nodes to execute the task using $AllocReq$ requests and $AllocConf$ confirmations, and returns the compound results from process $P_{Aggr}(data_{alloc})$ to the external PTD upon completion via the $RemRsrcRes$ message.

Application: *Alice allows Bob to run his image analysis software in her wearable cloud in exchange for micro-payments. Bob knows Alice and is aware of her physical location, and he feels comfortable that his personal data never leaves the proximity and is not uploaded to the public clouds.*

C. Phone Virtualization

Authorized and authenticated thin PTDs can utilize a virtualized phone obtained from a proximal wearable cloud, as illustrated in Figure 3(c). An external and proximal PTD can request for a phone virtualization service from a nearby wearable cloud using the $RemVPReq$. The wearable cloud owner is notified on the PDT by the CAP via the $NotfRemVPReq$ message. Upon successful authorization, the CAP allocates a VoIP channel for the external PTD using the elastic mini nodes via $AllocVoIP$ requests and $AllocVoIPConf$ confirmation. Subsequently, the external PTD can use a virtual phone service via $RemVoIPReq$ and encapsulated SIP messages [18].

Application: *Alice has a cellular subscription and uses her wearable cloud to use the service from multiple PTDs. She also trusts and allows Bob, who doesn't have cellular subscription, to use phone services using her wearable cloud in exchange for some payments. Bob trusts Alice and is visually aware of her physical proximity, and therefore, avoids the security and privacy issues of using a public cloud.*

D. Mobile and Wearable Service Composition

Multiple mobile and wearable devices can cooperate with their individual capabilities to provide a complex composite service via a unified handle using the wearable cloud, as illustrated in Figure 3(d). The CAP uses the service monitor $P_{ServMon}(cap_{dev})$ to evaluate individual devices' capabilities. Once a $ServReq$ request is received, the CAP requests the available devices for the service using $CompReq$ composition requests and $CompRes$ composition responses. The service compositor process $P_{Comp}(data_{comp})$ then aggregates the service, utilizing the elastic mini nodes via $AllocReq$ requests and $AllocConf$ allocation confirmations. Finally, the composite service session $CompSes$ is established within the cooperating devices.

Application: *Alice wears a health monitor wristband and views the analysis of the data on her smart-phone. When she returns home, her kitchen counter-top tablet screen displays the list of options for her dinner, based on what she has stored in her smart refrigerator. All computations are performed locally using her wearable cloud without the necessity of a provider-specific cloud.*

E. Delegated Mobile and Wearable Experience

Feature limited device owners can request a wearable cloud for a delegated service experience, as illustrated in Figure 3(e). The CAP receives a $DSEReq$ request. The CAP notifies the wearable cloud owner via a PTD using the $NotfDSEReq$ message. Once authorized, the external PTD sends a description of the delegated service experience $DSEDec$. The service monitor $P_{ServMon}(cap_{dev})$ at the CAP evaluates the request and determines the PTDs which are required for the current context using $CompReq$ requests and $CompRes$ responses, and allocates (if required) the necessary elastic nodes using $AllocReq$ requests and $AllocConf$ confirmations. The service compositor $P_{Comp}(data_{comp})$ then establishes the delegated composite service session $DSECompSes$ for the external PTD with the wearable cloud.

(a) Inside view **(b) Front view** **(c) Worn on body**

Fig. 4: Wearable cloud jacket using Raspberry Pis.

Application: *Alice owns some wearable devices and is able to enjoy a smart experience using the wearable cloud. Bob does not own any such device. Alice therefore allows Bob to experience some of the smart features via the wearable cloud to his terminal device in exchange for some micro-payments.*

V. Prototype Implementation

In this section, we present a proof-of-concept implementation and performance evaluation for the wearable cloud.

A. Prototype Components

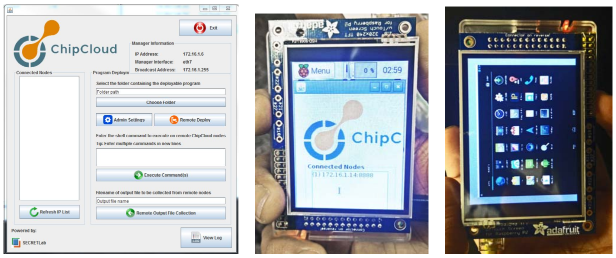
Cloud Jacket: We created a wearable cloud using a jacket and 8 Raspberry Pis (R-Pis). The R-Pis were networked over WiFi and powered by a portable power hub. We placed 4 R-Pis on each side of the front vest and were mounted with customized plastic casings. We utilized 5 R-Pi 2 Model B nodes, with 900MHz quad-core ARM Cortex-A7 CPU and 1 GB RAM, and 3 R-Pi Model B+ nodes, with 700MHz Broadcom BCM2835 CPU and 512 MB RAM. Figures 4a, 4b, and 4c show the cloud jacket that we created using the devices. The red boxes demark the placement of the R-Pis inside the side vests of the jacket.

Cloud Access Point: We created a Java-based managerial application, *ChipCloud*, for the wearable cloud, as illustrated in Figure 5a. *ChipCloud* allows easy deployment and execution of distributed tasks to the R-Pis in the wearable cloud via WiFi. The node coordinator is responsible for distributed parallel execution of the tasks and can remotely collect the results from the individual R-Pi nodes.

Cheap Terminal Device: We constructed a terminal device to imitate a cheap user-interactive device using an R-Pi and a 2.8 inch touchscreen, as shown in Figure 5b. The total cost of the device was only USD $64 (Raspberry Pi 2 Model B: $35 + 2.8 inch TFT touch screen: $29). The device was utilized to interact with *ChipCloud*. We also utilized the terminal to remotely access an Android smartphone dashboard via the wearable cloud, as shown in Figure 5c.

B. Performance Evaluation

Setup: We utilized *ChipCloud* to perform experimental evaluation for the wearable cloud architecture. We performed



(a) ChipCloud manager (b) ChipCloud on terminal device (c) Remote Android on terminal device

Fig. 5: Prototype application implementation of wearable cloud.

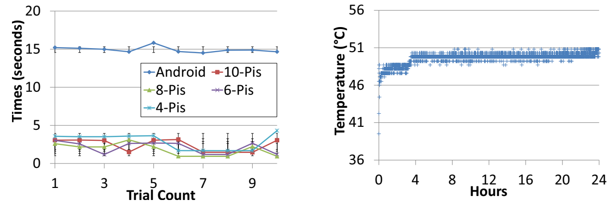
distributed task processing of the character count problem on 10 R-Pis, with 5 each of Pi-2 and Pi B+ models. We obtained a large text file of size 8.01 MB. The *ChipCloud* task allocator split the text into small chunks of 4, 6, 8, and 10 equal pieces and performed distributed processing using 4, 6, 8, and 10 R-Pis. We also executed the same program natively on an Android Nexus 4 mobile phone as a single file.

Time Measurement: As seen on Figure 6a, the Android phone took the longest time, with an average of 14939.9 ms. The distributed processing for the R-Pi nodes took lesser time. Each chunk of text file while using 4 nodes was the largest (appx. 2.00 MB) and took 2884.8 ms to complete. The lowest time was taken while using 8 nodes for the task (1822.1 ms). The performance for R-Pis fluctuated because of the network latency and the number of parallel sessions, and the required time increased to 2428.3 ms in the case of 10 nodes.

Power and Heat: We executed the character count problem over SSH server via *ChipCloud* continuously for 24 hours and measured the system temperature on an R-Pi 2 at 30 second intervals using the on-board temperature sensor. Figure 6b presents the recorded temperature of the R-Pi. The mean and maximum operating temperatures were 49.67°C (121.4°F) and 51.0°C (123.8°F) respectively, with the room temperature fixed at 23.33°C (74.0°F). The exterior temperature of the plastic casing (measured using an external thermometer) was always found to be at room temperature. The R-Pi 2 and B+ have comparable power usage and were running from a USB power distribution hub with 5V/950mA maximal power rating. The R-Pis used approximately 200mA while idle and upto a variable maximum of 350mA while processing.

VI. Related Work

The Smart Vest [7] is a wearable computer with chip based components. However, our proposed wearable cloud does not focus on the fabrication of the components, rather, the allocation of elastic mini chip based computers to allow a pervasive experience for mobile/wearable device users. mCloud is a mobile device based ad hoc cloud where mobile phones form a cloud computing platform [15]. Hasan et al. have proposed Aura to offload computation from mobile devices to Internet-of-Things devices [16]. In our case, we propose offloading the tasks to the personal wearable cloud, and thus, making it more secure with the data in immediate proximity. Hoang et al. [5] described a mobile cloud architecture using sensors and mobile nodes to collect and manage data from the environment. We intend to provide a generalized service where the computation is done via portable mini computing devices, where the mobile



(a) Time taken for character count (b) System temperature

Fig. 6: Prototype evaluation measurements

phones are the clients rather than cloud nodes. Cloudlet-based models [8] are designed as widely available zones in the public space to provide computational resources. Unlike our proposed wearable cloud architecture, such models complicate the security aspects of offloading schemes and data privacy. Noor et al. [17] presented CellCloud, a mobile cloud built using mobile phones where the cellular base station acts as the controlling node, which can be leveraged to distribute the tasks within the mini nodes in the wearable cloud framework.

Khan et al. proposed an interaction provenance based scheme [19, 20] for interactive services. The scheme can be applied towards easier service composition in a secure and privacy-preserving manner. Various research have also been proposed in terms of secure and distributed entity and identity management techniques for ad hoc, cooperative, and evolving architectures [21, 22]. The proposed wearable cloud can benefit using efficient task estimation methods, as proposed by Kosta et al. [23] and Lee et al. [24], and can be adopted for task allocation in the wearable cloud. Heuristic-based approaches can also be effectively adopted to allocate the nodes in the wearable cloud [25].

Researchers have explored the problem of mobile computation offloading [26–29]. Task offloading models primarily deal with offloading computation requests to another public entity [30] and allow prediction based offloading for mobile nodes with intermittent connectivity [31]. Our wearable cloud allows task requests in a private environment with guaranteed proximity and availability. An elastic application partitioning model is proposed in [32] to augment the computation capabilities of mobile devices. The wearable cloud can also adopt service oriented models, as proposed by Shi et al. [33], for computational offloading from the terminal devices to the cloud access point. A mobile cloud architecture is proposed in [34] in order to asynchronously delegate resource-intensive mobile tasks to alleviate the mobile device load.

Adaptive probabilistic scheduler for offloading time-constrained or real time tasks in local mobile clouds is proposed in [35]. A mobile device cloud platform is proposed in [36], that enables the creation and assessment of various offloading algorithms, in order to have a significant reduction in computation time and energy consumption. To overcome the limited resources of mobile platforms, a machine learning-based runtime scheduler for task offloading to external powerful resources, such as, personal work-stations or cloud servers, was discussed in [37]. Our work is complementary to the research on computation offloading and can leverage the offloading scheme or scheduling algorithms to determine optimal strategies for offloading tasks in the wearable cloud.

The earlier works can be utilized to enhance the operational efficiency of the wearable cloud.

VII. Discussion and Conclusion

The proposed architecture is a portable cloud which a user carries around along with other mobile and wearable devices. The wearable cloud allows smart utilization of energy, enhanced and cheap wearable and mobile devices, and a simpler security paradigm. We can also envision the creation of a hyper-wearable high-performance cloud in disaster and first-response scenarios via multiple cooperating wearable clouds. The wearable cloud allocates resources as per requirements to conserve maximum energy without compromising the user experience. The proposed model is different than using a laptop, given the static engagement of a system in portable computers, versus, the dynamic and requirement-based engagement of resources in the wearable cloud.

The recorded heat dissipation from the system provides reasonable justification for the suitability of chip-based computers for the wearable cloud. Even with an operating temperature of 51°C (max), the small form factor of the nodes results in minimal thermal energy dissipation. The exterior temperature on the casing for the nodes were the same as the room temperature, preventing any heat sensation for the jacket wearer. Moreover, the prototype components did not include any form of heat sink, which, we believe, will improve the cooling of the mini nodes. A commercial design for the wearable cloud jacket may also include an aluminum lining, thus making the whole jacket as a massive heat sink.

The prototype was powered using portable power banks. However, there are multiple cheap alternatives for the power supply, including off-the-shelf portable solar chargers[2a], motion-based portable chargers[2b], and self-charging power units [38], with upto 6000mAh capacity for as low as $10, enabling a single mini node to run upto 20 hours on a single charge in practical usage scenarios.

The cloud access point did not employ any intelligent node allocation strategies for the given prototype. The proposed architecture is a generalized overview of a possible form of mobile wearable cloud technology. There are many issues to be addressed in such form computing, such as task distribution strategies, inter-node and intra-node communication security, and service composition models using multiple devices. We are currently working on the possibility of a generalized framework for seamless service integration using multiple devices and cooperative applications. Our future work includes the development of a comprehensive task distribution strategy by the wearable cloud access point.

Acknowledgements

This research was supported by the National Science Foundation CAREER Award CNS-1351038.

References

[1] "U.S. Smartphone use in 2015," Online at http://www.pewinternet.org/files/2015/03/PI_Smartphones_0401151.pdf, April 2015.

[2] R. Rettner, "Tracker craze: Fitness wristbands' popularity will continue to grow," Online at http://www.livescience.com/42144-activity-monitors-popularity.html, Dec 2013.

[3] "Raspberry pi model specifications," Online at https://www.raspberrypi.org/documentation/hardware/raspberrypi/models/specs.md.

[4] G. Fortino, M. Pathan, and G. Di Fatta, "Bodycloud: Integration of cloud computing and body sensor networks," in *Proc. of CloudCom*. IEEE, 2012.

[5] D. Hoang and L. Chen, "Mobile cloud for assistive healthcare (mocash)," in *Proc of APSCC*. IEEE, 2010.

[6] C. Doukas and I. Maglogiannis, "Bringing iot and cloud computing towards pervasive healthcare," in *Proc. of IMIS*. IEEE, 2012.

[7] S. J. Schwartz and A. Pentland, "The smart vest: towards a next generation wearable computing platform," 1999, mIT Media Laboratory Perceptual Computing Section Technical Report No. 504.

[8] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14–23, Oct 2009.

[9] "LG G Watch," Online at https://www.qualcomm.com/products/snapdragon/wearables/lg-g-watch.

[10] "Android benchmarks," Online at https://browser.primatelabs.com/android-benchmarks.

[11] "Mac benchmarks," Online at http://browser.primatelabs.com/mac-benchmarks.

[12] "Google glass review," Online at http://www.techradar.com/us/reviews/gadgets/google-glass-1152283/review/7.

[13] "Teardown Puts iPhone 6 Cost at $200," Online at http://www.pcmag.com/article2/0,2817,2469089,00.asp.

[14] "Apple iPhone 6," Online at http://store.apple.com/us/buy-iphone/iphone6/4.7-inch-display-16gb-space-gray.

[15] E. Miluzzo, R. Cáceres, and Y.-F. Chen, "Vision: mClouds-computing on clouds of mobile devices," in *Proc. of MobiSys*. ACM, 2012.

[16] R. Hasan, M. M. Hossain, and R. Khan, "Aura: An iot based cloud infrastructure for localized mobile computation outsourcing," in *Proc. of Mobile Cloud*. IEEE, 2015.

[17] S. Noor, M. Haque, and R. Hasan, "Cellcloud: a novel cost effective formation of mobile cloud based on bidding incentives," in *Proc. of Cloud*. IEEE, June 2014.

[18] H. Sinnreich and A. B. Johnston, *Internet communications using SIP: delivering VoIP and multimedia services with Session Initiation Protocol*. New York, USA: John Wiley & Sons, Inc., ISBN 0-471-41399-2, 2001.

[19] R. Hasan and R. Khan, "Interaction provenance model for unified authentication factors in service oriented computing," in *Proc. of CODASPY*. IEEE, 2014.

[20] R. Khan and R. Hasan, "Fuzzy authentication using interaction provenance in service oriented computing," in *Proc. of SCC*. IEEE, 2015.

[21] R. Khan and R. Hasan, "MIDEP: Multiparty Identity Establishment Protocol for Decentralized Collaborative Services," in *Proc. of SCC*. IEEE, 2015.

[22] R. Khan and R. Hasan, "SecP2PSIP: A Distributed Overlay Architecture for Secure P2PSIP," in *Proc. of Cyber Security Conf.* ASE, 2014.

[23] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. of INFOCOM*. IEEE, 2012.

[24] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *The Journal of Supercomputing*, vol. 60, no. 2, pp. 268–280, 2012.

[25] B. Li, Y. Pei, H. Wu, and B. Shen, "Heuristics to allocate high-performance cloudlets for computation offloading in mobile ad hoc clouds," *The Journal of Supercomputing*, pp. 1–28, 2015.

[26] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1294–1313, 2013.

[27] X. Ma, Y. Cui, L. Wang, and I. Stojmenovic, "Energy optimizations for mobile terminals via computation offloading," in *Proc. of PDGC*. IEEE, 2012.

[28] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: making smartphones last longer with code offload," in *Proc. of MobiSys*. ACM, 2010.

[29] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proc. of EuroSys*. ACM, 2011.

[30] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE TPDS*, vol. 26, no. 4, pp. 974–983, April 2015.

[31] B. Li, Z. Liu, Y. Pei, and H. Wu, "Mobility prediction based opportunistic computational offloading for mobile device cloud," in *Proc. of CSE*. IEEE, Dec 2014, pp. 786–792.

[32] X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, "Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing," *Mobile Networks and Applications*, 2011.

[33] C. Shi, K. Habak, P. Pandurangan, M. Ammar, M. Naik, and E. Zegura, "Cosmos: Computation offloading as a service for mobile devices," in *Proc. of MobiHoc*. ACM, 2014.

[34] T. Justino and R. Buyya, "Outsourcing resource-intensive tasks from mobile apps to clouds: Android and aneka integration," in *Proc. of CCEM*. IEEE, 2014.

[35] T. Shi, M. Yang, Y. Jiang, X. Li, and Q. Lei, "An adaptive probabilistic scheduler for offloading time-constrained tasks in local mobile clouds," in *Proc. of ICUFN*. IEEE, 2014.

[36] A. Mtibaa, K. A. Harras, and A. Fahim, "Towards computational fffloading in mobile device clouds," in *Proc. of CloudCom*. IEEE, 2013.

[37] H. Eom, P. S. Juste, R. Figueiredo, O. Tickoo, R. Illikkal, and R. Iyer, "Machine learning-based runtime scheduler for mobile offloading framework," in *Proc. of CloudCom*. IEEE, 2013.

[38] X. Pu, L. Li, H. Song, C. Du, Z. Zhao, C. Jiang, G. Cao, W. Hu, and Z. L. Wang, "A self-charging power unit by integration of a textile triboelectric nanogenerator and a flexible lithium-ion battery for wearable electronics," *Advanced Materials*, vol. 27, no. 15, pp. 2472–2478, 2015.

[2](a) www.everbuying.net/product1101363.html, (b) www.getampy.com