Performance Evaluation of Wearable Computing Frameworks

Michelle Cacais, Danielo G. Gomes,
Paulo Armando C. Aguilar, Rossana M. C. Andrade

[1]Universidade Federal do Ceará (UFC)
Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)
Av. Mister Hull, s/n – Campus do Pici – Bloco 942-A
60455-760 – Fortaleza – CE – Brazil

{cacais, dgomes, paulo, rossana}@great.ufc.br

Abstract. *Internet of Things (IoT) aims to connect multiple devices and enable communication on a global scale. In this context, wearable computing turns into one of the most imperative technologies for data capture and transfer. Even though many solutions exist for fast development of wearable applications, it is difficult to know which one is the best for each situation. In this paper, we evaluated the performance of three wearable computing frameworks: SPINE, BNSM and IBM Bluemix. We used performance metrics such as data accuracy, memory footprint and energy consumption in a movement recognition scenario. Results have shown that, among the analyzed frameworks, (i) BNSM is the best one concerning energy-efficiency; (ii) IBM Bluemix is the best one for data accuracy. Both frameworks presented interesting results in terms of memory footprint.*

1. Introduction

In the context of Internet of Things (IoT), a lot of devices have been changed into smart objects to perform tasks in human routine. The idea of access information everywhere as well the low cost of body sensors have motivated the creation of wearable systems related to many fields, such as healthcare, e-sports, e-fitness, entertainment and interactive games. Wearable computing is one of the IoT core technologies, in which ”things” (e.g. a body sensor) interact with each other to share information and synchronize processes. Wearable computing also brings a way of staying connected and being constantly monitoring through the garment or other accessory attached to the body.

Wearable computing is useful to collect and provide data about people activity and behavior. However, developing artifacts using wearable technology is a difficult task due to the challenge of managing body sensors, embedded systems and network protocols. The developers have to know about software, hardware and sensor networks to create functional wearable systems. As a consequence, a wearable device becomes complex and hard to reuse. In addition, it is difficult to choose a certain framework for fast prototyping and development of wearable appliances. Besides, it is still not easy to find surveys or studies related to the performance of the current wearable approaches, what makes it harder to find support to develop wearable systems.

Here we propose a systematic performance evaluation of three selected wearable computing frameworks by using system metrics such as memory and energy consumption, bandwidth and data accuracy. The evaluated frameworks have been chosen from criteria such as code availability, relevance, freshness and general use. We used a real body sensor for motion capture. Furthermore, we take into account some qualitative aspects of the frameworks such as easy of development, debugging and reuse.

2. IoT and Wearable Computing

The term Internet of Things refers to the idea of everyday objects aware of the context and able to collect and exchange data using the Internet. This technology supports the connection of real-world devices to sensors, microprocessors and actuators to send, receive and process many types of data, such as temperature, humidity, movement, and other variables [Borgia et al. 2016]. In this context, the range of applications that can be created is huge, involving the real and the virtual to create smart environments by modifying energy consumption, transport, and even entire cities with artifacts interacting and cooperating. The goal is to connect smart objects anywhere, anytime, using any network and any service.

IoT gives support to deploy smarter cities, environmental monitoring and urban computing, automotive and wearable computing. The last one is related to devices worn to the body, constantly on, non-monopolized, aware of the context and able to communicate with the user, allowing its control [Hussein 2015]. Wearable computing brings the advantages of pervasive computing by placing sensory resources in a simple way, like embedded devices in clothes, for example. This way, wearable offers new models of interaction with the user, unlike smartphones or tablets that are designed for hand-held use. Because of this feature, wearable is often designed to be heads-up and speech activated.

Wearable computing also brings some challenges, as energy consumption, heterogeneity of devices and memory footprint. The development of wearable infrastructures involves different hardware, what means dissimilar accuracy, frequency and transmission rate. Data accuracy is also a challenge of pre-processing. This procedure involves noise removal, motion artifacts, data formatting, normalization and synchronization. In addiction, body sensors are rich sources of data, so the storage system has to be capable to support the wearable workload or has to be adapted to dynamically manage limited resources. The developer has to keep in mind that it will be faced these challenges with a wearable infrastructure.

3. Material and methods

In this paper, we analyzed three wearable computing frameworks to evaluate their performance. For a fair evaluation, we implemented an application for motion recognition using each framework as a case study. The main goal of this approach is to check how these wearable solutions behave to solve some resource constraints, such as memory footprint, energy consumption and data accuracy. We measured and simulated.

3.1. Selected frameworks

There are a lot of solutions for wearable computing in the literature. To select and chose the wearable frameworks to be evaluated, we made a sort of simple systematic review using search strings as ”wearable computing framework”, ”wearable computing rapid development” and ”body sensor networks framework”. The research generated a large number of works, including frameworks, architectures and libraries faced to many proposes, mainly in the area of health care. To limit this number, it has been chosen titles more generic, it means, those that were not concerned with solving only specific problems.

Also, It has been taken into account availability of the code, freshness and relevance. At the end, three frameworks have been chosen: SPINE [Bellifemine et al. 2011], BNSM [Iyengar 2009] and IBM Bluemix [Gheith et al. 2016].

3.1.1. SPINE

The framework named Signal Processing In Node Environment (SPINE) [Bellifemine et al. 2011] is a domain-specific tool for rapid prototyping of WBSN applications. SPINE is an open source project that gives support to local and remote applications with lightweight Java APIs, including protocols, utilities and data processing functions. It can be used to manage sensor nodes or issue service requests. It also allows the rapid prototyping of distributed Body Sensor Networks (BSN) algorithms for data classification. SPINE works with multiple sensor nodes and one coordinator, which acts like a gateway and manages the networks, the collection and analysis of data.

3.1.2. BNSM

The work described in [Iyengar 2009] presents the framework called Body Network Service Manager (BNSM), an appliance that focuses on defining abstractions to allow researchers of different areas to develop using BSNs. It creates an abstraction layer between the application and the nodes named Network Service Manager (NSM), which coordinates the network and responds by generating events. The framework can trigger many events asynchronously, as arrival of data, network status updates and error notifications. The NSM includes the discovery of nodes in the network, processing of requests, disseminating of data and query network status. The nodes can perform tasks as send or receive packets, sample sensors, store data and perform computation, all of these controlled by managers.

3.1.3. IBM Bluemix

Bluemix is the cloud offered by IBM [Gheith et al. 2016]. It consists in a platform for cloud-based mobile applications. IBM provides services as file storage, application authentication, push notifications, and server-side application logic. It is possible to access these through easy-to-use client software development kits (SDKs) available for download. This framework facilitates the creation of wearable devices using cloud services, facilitating the connection and extensibility. The cloud is based on Platform as a Service (PaaS) and it is private. The server-side of IBM Bluemix was implemented with components as stateless services for scalability and fault resilience.

3.2. Case study: human activity recognition

In this case study, each framework have been used to design an application for Android devices to recognize and classify human movement. Literature includes ways to classify gestures with camera and gravity sensors. In this experiment, the sensor used was the accelerometer, a hardware that measure the acceleration force in m/s_2, applied to a device on three physical axes (x, y and z), including gravity. The motions are classified into sitting, standing, walking and falling for the measurements. The application has to receive data from the sensor and identify in which category it is classified. The goal with this case study is to verify how the frameworks simplify implementation in a wearable context.

4. Performance Evaluation

We made real measurements using a device and performing the actions for recognition. The goal of this task was to verify how a real application behave in a real context using the frameworks. It had been made three Android apps using each framework. The tests were made using an actual measurement with a smartphone attached to the belt of a person and the actions were performed for 20 times each one. The results was obtained with series of movements made of a person with a smartphone. Each movement was executed 20 times, obtaining 80 samples in total. The time for starting the movement until the end of recognition took about 35 seconds, and the average time to complete one action was 12 minutes for each type of movement.

The system parameters include a smartphone called Samsung J1, with a 1230 mAh battery, a processing power of Dual-Core 1.2 GHz and operating system Android 5.0 (Lollipop). The server side was implemented in Java Language for both SPINE and BNSM, and for IBM Bluemix, it was used the cloud of the company, with no need of install or configuration. The workload parameters include the number of data (20 samples for each movement), number of possible movements (4), time in seconds necessary to complete one movement and total time of execution.

4.0.1. Results

The consumption of energy has been measured for all cases using the frameworks. Figure 5 shows the consumption of battery used to capture one movement (about 35 seconds). It is possible to note that BNSM has had the best when compared to the other ones. This happened because of the functions and events management of the framework. The solution provided by IBM also had good results. SPINE was the framework that consumed more battery. It may occurred because of the centralized architecture of this one.

The consumption of memory is a problem of allocation. Figure 4.0.1 shows the results of the tests using the case study. This procedure measured the size occupied of the random-access memory (RAM) of the smartphone. Both frameworks had similar results, but it is possible to see that the solution of IBM Bluemix consumed less memory compared to the others. SPINE had better results than BNSM because of its dynamic-memory management.

Table 1 shows the result of accuracy of the frameworks. That depends of the algorithm used in each framework. In order to obtain reliable data, it was used the solutions provided for them in their work. SPINE uses a motion recognition algorithm called K-Nearest Neighbor (KNN). For BNSM, it was utilized the model of classification that the authors provided. IBM also provided an application for motion recognition. At the end of the tests, all frameworks had good results, but IBM was more accurate.

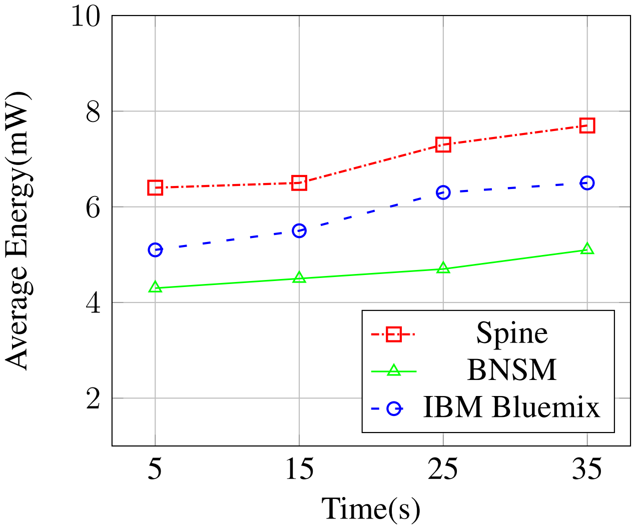

Figure 1. Energy consumption (real measurements).

Table 1. Accuracy of real data

	Sitting	Standing	Walking	Falling
SPINE	95%	93%	93%	99%
BNSM	95%	95%	96%	100%
IBM	96%	97%	98%	100%

4.1. Analyses

The usage of frameworks involves abstraction of the code, and the main goal of using them is to facilitate development of computing projects and rapid prototypes. The frameworks analyzed in this experiment meet this expectation, because they provide resources that minimize the need of generate code of the server-side, the nodes and the communication between sensors, gateway and cloud. However, it is not a trivial task to configure them. There is the need of correct some errors, unavoidable when dealing with portability. In this context IBM Bluemix solution is the easier to use.

The results of the real measurements have shown that BNSM is the most appropriated for energy-efficiency and IBM Bluemix is the one that had better results with accuracy and memory footprint, in a context of a simple application of movement recognition. The simulation had similar results compared to the real measurement, what make more reliable the data obtained in this procedure. It is possible to conclude that both frameworks obtained good results with this parameters analyzed. Table 2 shows the results with the confidence interval, estimated by the number of repetitions and samples.

Table 2. Average of the results with a confidence interval

	Energy consumption (mW/s)	Memory footprint (KB)	Accuracy (%)
SPINE	6.9 ± 1.7	4.2 ± 0.8	93 ± 1
BNSM	4.8 ± 1.7	4.5 ± 0.8	95 ± 1
IBM	5.8 ± 1.7	3.2 ± 0.8	97 ± 1

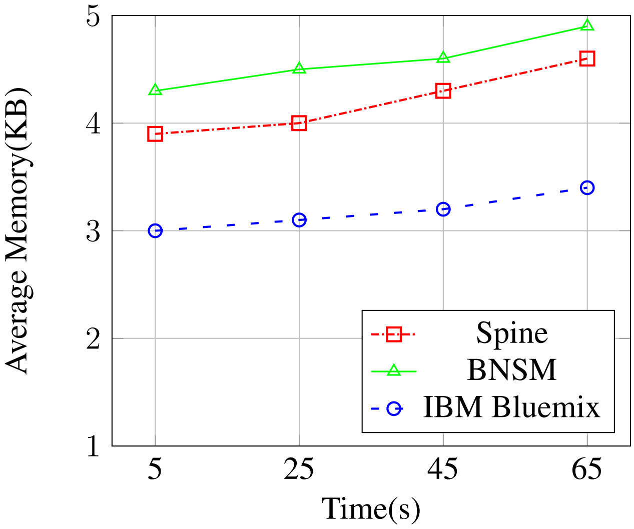

Figure 2. Memory footprint (real measurements).

5. Conclusion

The frameworks evaluated in this paper showed as good options to be used in wearable development. All of them help with heterogeneity of protocols and nodes. Also, all frameworks provide good documentation and examples of applications using the code available. The solution provided by IBM Bluemix was the easiest to use for rapid prototyping, but it is not possible to have access to the server-side, what limits the possibilities of development and the control over the applications. SPINE and BNSM have shown similar results, but the second one was more efficient for energy saving.

Acknowledgments

Michelle Cacais received financial support due scholarship from CAPES (process #1589313). Danielo G. Gomes and Rossana M. C. Andrade are CNPq Fellows.

References

Bellifemine, F., Fortino, G., Giannantonio, R., Gravina, R., Guerrieri, A., and Sgroi, M. (2011). Spine: a domain-specific framework for rapid prototyping of wbsn applications. *Software: Practice and Experience*, 41(3):237–265.

Borgia, E., Gomes, D. G., Lagesse, B., Lea, R., and Puccinelli, D. (2016). Special issue on "internet of things: Research challenges and solutions. *Computer Communications*, 89–90:1 – 4.

Gheith, A., Rajamony, R., Bohrer, P., Agarwal, K., Kistler, M., Eagle, B. W., Hambridge, C., Carter, J., and Kaplinger, T. (2016). Ibm bluemix mobile cloud services. *IBM Journal of Research and Development*, 60(2-3):7–1.

Hussein, A. I. (2015). Wearable computing: Challenges of implementation and its future. In *12th Learning and Technology Conference, 2015. 35752 2015.*, pages 14–19. IEEE.

Iyengar, S. (2009). Framework for body sensor networks. Master's thesis, EECS Department, University of California, Berkeley.