Peramanathan Sathyamoorthy

Enabling Energy Efficiency Data Communications with Participatory Sensing and Mobile Cloud

Cloud-assisted crowd-sourced data-driven optimization

UPPSALA
UNIVERSITET

Abstract

This paper proposes a novel power management solution for the resource-constrained devices in the context of Internet of Things (IoT). We focus on smart-phones in the IoT, as they are getting increasingly popular and equipped with strong sensing capabilities. Smart-phones have complex and chaotic asynchronous power consumption incurred by heterogeneous components including their on-board sensors. Their interaction with the cloud can support computation offloading and remote data access via the network. In this work, we aim at monitoring the power consumption behaviours of smart-phones and profiling individual applications and platform to make better decisions in power management. A solution is to design architecture of cloud orchestration as an epic predictor of the behaviours of smart devices with respect to time, location, and context. We design and implement this architecture to provide an integrated cloud-based energy monitoring service. This service enables the monitoring of power consumption on smart-phones and support data analysis on massive data logs collected by large number of users.

Dedicated to
The eyes: Wife and Mom
The shoulders: Dad and Father-in-law
My brother Thileepan
Two little stars : Charu Nethra Priyanga
&
To all my great supporting friends

List of papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.

I A Paper Discussed in this Thesis

Reprints were made with permission from the publishers.

Contents

1. Introduction

IoT is a convergence of number of technologies such as *sensors*, *IPv6*, *wireless communication* and *Internet*. Any real-world objects become smart just by satisfying few conditions but not limited to: 1) uniquely identifiable; 2) being able to sense or actuate; 3) being able to communicate [**?**]. The growth of smart objects are posing challenges to the research community in energy management, data analytics and security [**?**]. Among these challenges, security and privacy issues are not just issues in technical system design level, but also in ethical, behavioural and policies level. We have powerful analytical tools available with advanced data analysis algorithms [**?**]. On the other hand, energy management is more complex and chaotic, that is our focus of this paper.

Berkeley National Laboratory defined energy efficiency as using less energy to provide the same service. The need for energy efficiency highly inevitable in almost every type of industries, companies and organizations including Information and Communications Technology (ICT). Energy management in Internet of Things(IoT) aims at reducing the electricity, which is beneficial for many industries to reduce their electricity bills. As the smart objects becoming smaller in size, their small sized batteries provide limited power for operations. Even the smart appliances are idle, they could indirectly waste huge amount of energy in long term and eventually increase the electricity bills too. "Although ICT can enable energy efficiency across all sectors, at present there is little market incentive to ensure that network-enabled devices themselves are energy efficient. In fact, up to 80% of their electricity consumption is used just to maintain a network connection. While the quantity of electricity used by each device is small, the anticipated massive deployment and widespread use makes the cumulative consumption considerable" as reported by International Energy Agency in [**?**].

Hereafter we narrow our focus on smart-phones which are increasing in exponential order over the years. Modern smart-phones provide heterogeneous functionalities including a number of sensors. They are one of the most representative and popular smart objects in the IoT. As smart-phones are resource constraint with respect to memory and computation, they happen to off-load computation and access remote storage on the cloud servers via network. Cloud computing in the IoT leads to thousands of cloud supported applications and it is growing steeply. As a consequence, smart-phones are consuming a lot of energy for communication with the cloud. Due to the size limitation, effort of making powerful batteries is not able to withstand the energy hungriness persist in the smart-phones. It is important to reduce energy consumption when developing new kind of applications.

Smart-phones are usually running multiple applications with different operations at the same time. It is very difficult to understand and identify the cause of high energy consumption in this asynchronous power consuming environment. It is necessary to provide profiling of power consumption in at different levels, such as whole system, individual applications, and system calls in operation level. In this paper, we propose the first iterative novel solution using *Cloud Orchestration* for power management on smart-phones. Cloud orchestration aggregates power profiling data from the smart-phones and coordinates data storage, data analysis, learning and decision making. From the profiling data, the orchestrator learns mainly the power consumption behaviours and the usage pattern of the participating smart-phones. The orchestrator aims for providing overall system power management rather than making part of the system efficient.

2. Background

There has been many efforts made to enable energy efficiency in smart-phones and in IoT in general. They are a range of solutions tried out in *Hardware Architecture* level [**?**, **?**], *Data communication* level [**?**, **?**], *Network infrastructure* level [**?**] and in *Protocols* optimization [**?**]. As Intel summed up in [**?**], *Software Energy Efficiency* is towards achieving *Computational Efficiency*, *Data Efficiency*, *Context Awareness* and *Idle Efficiency* in broader sense. Few common problems with most of the existing solutions are including: 1) system as a whole was not considered; 2) trade-off between components was not properly considered; 3) interdependences of the components was not properly studied; 4) the existing solutions are suboptimal. For measuring energy consumption, solid background has been provided in [**?**]. Internet of Things-Architecture, a consortium is rigorously developing architectural reference models. The models could potentially serve the best initial guidance towards concrete architecture for the problem of interest and eventually towards the actual system architecture [**?**]. In [**?**], devices orchestration is explained in business process point of view.

3. Orchestration

3.1 General Context

IoT initially had two visions, one is *Internet oriented* vision and another one is *Things oriented* vision. Later when new challenges introduced such as unique addressing and storing information, *Semantic oriented* vision had arisen [**?**]. According to the orientation, the participating devices are categorised and the orchestration is configured. In smart-phones, all the visions are exists.

3.1.1 IoT elements of smartphones

In [**?**], the general architecture of IoT is well explained as four layers. *Perception Layer* is where all the information collected, say, from the environment using specialised equipments such as sensors and GPS. Thus this layer provides the digital representation of physical world. *Network Layer* is for communication and transmission of information. *Middle-ware Layer* by its various intelligent cloud computing supports the application layer and act as a intermediate between network layer and application layer. *Application Layer* is fully customized with respect to the users and the purpose of the application. In some scenarios, both the perception layer and application layer reside in the same smart-phone system.

3.1.2 Need for orchestration

The main goal is to provide energy efficient decision(s) back to the service enabled smart-phones which are participating in the Orchestration. Orchestration, the concept existing in the music world was adopted in the process automation of business world by automating, coordinating and managing complex systems, middle-wares and services. In the context of vulnerability to energy efficiency, even a single piece of code (`while(battery.percentage) println(battery.percentage)`) could be potentially dangerous, leading any effort made to find suboptimal solutions into vain. Hence, there is a need for the great intelligent system, which is capable of finding and categorizing the energy errors. The system should have access to powerful *dynamic control system engine* for fixing such errors. To assist bug fixing we may need a strong insights from the big data of crowd sourced logs/operations over the time. The big data may lead to *side-effect services* (new enabling services other than energy-efficiency). Hence the energy and effort of making this grant system has huge benefits in return. Thus the *Orchestration* with the capabilities of integrating different types of clouds, processes and services is an suitable choice.

3.1.3 Composition of systems and models and services

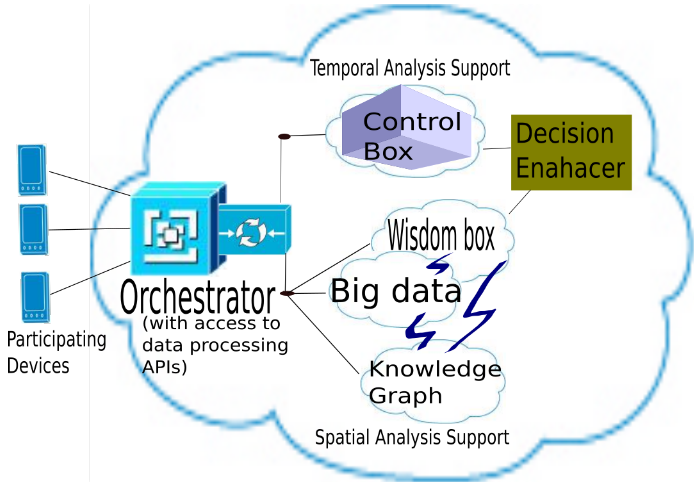


Figure 3.1. EEaaS Orchestration

We propose an architecture design of orchestrator as depicted in Figure 3.1, with the primary understanding of the participating systems. Then to evaluate and enhance, any existing reference models or relevant solutions in the other related domains may be adopted in the future. The design is open and flexible which make it possible to add,remove and merge number of *systems, models and services* at any granular level in the orchestrator. The orchestrator organizes the following components, *Participating Devices*, *Data Processor*, *Big data storage*, *Knowledge graph*, *Wisdom box*, *Control box* and *Decision Enhancer*

Participating devices
These are the devices interested in optimized energy usage. Upon registration with orchestrator, a dark skinned, fully customizable, lower energy consuming background *service application* is enabled in the devices. This application sends low level system-call logs periodically and reports strange system behaviours spontaneously. To avoid security and privacy issues, logs are collected anonymously with unique device profile. This application not only a logs collector, it will also act as local *self-controller*, try to catch energy errors in-time and its functionalities are regularly updated by orchestrator. This is to avoid the need of continuous data communication.

Data processor

Data processor is a collection of APIs for various data processing methods accessible to orchestrator. According to the context, methods will be chosen by the orchestrator. Here the data is processed for both big data analysis and temporal analysis.

Big data storage and modern tools

The data produced by smart-phones would be in massive scale over the time. Thus to handle this data-intensiveness we require big data storage and modern cloud programming paradigms such as *Hadoop* and *Apache flink*. For deep analysis of sample data, powerful computing languages such as *Python* and *R* are required.

Knowledge graph

By referring the big data of logs, dynamic knowledge graph is built and then keep on updated. Nodes are qualified classes and subclasses with attribute-value pairs. So it provides a clear and structured view of data. Using this graph, it is then easy to get specific data for analysis with respect to location, device model, internet service provider and for various specification.

Wisdom box

Wisdom box contains set of learners whose primary focus is building location specific insights (spatial domain). The box act as a predictor of trends in data, usage patterns, system behaviour anomalies. It uses combination of statistical algorithms and machine learning algorithms to find energy efficient decisions. The decisions that are independent of device, platform and applications is stored in *Decision Enhancer* in the orchestration. Device, platform and applications specific decisions fused in the knowledge graph and the reference graph stored in the decision enhancer.

Control box

Control box is a builder of real-time self-controllers for the participating *dynamic systems* with the help of time-sensitive feedbacks (temporal domain). These self-controllers embedded as a service as explained before in 3.1.3. To make these self-controllers even better, context related decisions in spatial domain are used. Feedbacks are received from the participating devices to evaluate the performance along with log collection.

4. Results

Smart-phone is a system-on-chip architecture with three key components *Application processor* to handle user applications, *Modem processor* to handle transmission and reception and *Peripheral devices(I/O)* to interact with users. In smart-phones, power consumption of any I/O component is sometime higher than the power consumption of the CPU or at the least comparable. In [**?**] the problems and flaws in the power models derived from external power meters and those derived from (hardware) utilization-based well explained by considering the types of components having *tail power states*, not having quantitative utilization(e.g. camera on/off) and system calls that change power states but not mean any utilization of hardware. As system calls are the only way for application can use I/O components, system call based power modelling is better one. Thus Energy profiling with all its overheads is the very important first step.

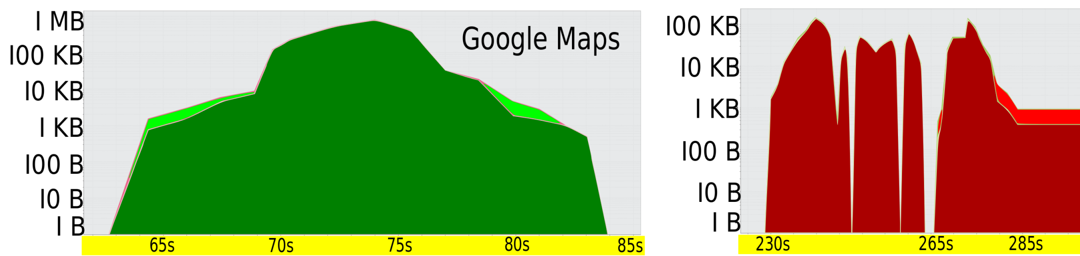
4.0.1 Power states,logging and semantic data

The Advanced Configuration and Power Interface(ACPI) specification has been evolving as a de-facto common hardware interfaces in Operating System-directed configuration and Power Management(OSPM) for both devices and entire systems. When profiling either an individual application or entire platform it is useful to fetch information about the states of system, device and processors, as in table 4.1, so that better decision can be made to achieve energy efficiency. We are currently using Qualcomm's Trepn Profiler [**?**]. We study the behaviours of the system and applications running in the system including CPUs usage, memory usage, data usage.

Global System States	Device Power States	Processor Power States
G0 Working	D0 - Fully-On	C0
G1 Sleeping	D1	C1
G2/S5 Soft Off	D2	C2
G3 Mechanical Off	D3hot	C3
	D3 - Off	

Table 4.1. *ACPI/OSPM defined power states*

As data communication is one of the main reasons for the quicker energy drain in the smart-phones, it is interesting example to profile the data usage patterns of the different applications. In figure 4.1, profiled data usage patterns



(a) *Google maps* **(b)** *YouTube*

Figure 4.1. Randomly profiled data usage patterns

of two applications, namely, *Google Maps* and *YouTube* is shown. While profiling google maps, GPS was turned ON at 20th second, maps were zoomed-in and then the device is navigated in random direction. We observed as shown in figure 4.1a, sudden high usage of data occurred about 10 seconds in the time interval [70, 80]. In this manual profiling and inspection, it is inferred that action of zooming-in the map is the reason for high data usage. Does the state of the GPS being ON has any effect on data usage? The answer seems *no* because of the fact that the GPS is turned ON at 20th second already. If the device lay in the fixed position, the above question doesn't make sense. If the device is in navigation, then it requires fine-tuned profiling again. While profiling youtube, a video is randomly picked for playing, after few seconds the screen is rotated to play the video in full screen mode, the video is continued to play. We observed as shown in figure 4.1b, the initial aggressive data communications due to pre-fetching. These scenario implies:

- Usage of resources such as CPU, Memory, GPU and Data varies with respect to applications
- Weighted effect on energy usage with respected to the resources should be studied
- Fine-grained profiling is data-intensive and computations-intensive
- Profiler should be self-energy efficient
- Automation of profiling with added semantics is extremely challenging
- Detailed study of interdependencies and contexts is required
- Mobile cloud computing is suitable

We are developing R computing language configured cloud application *EnergyApp* to develop and test energy models and visualize the insights on sample data. The application right now manually let the user to inspect the data collected by Trepn profiler. It also shows simple interactive plots where one can compare the parameters for example, as CPU usage against memory usage for the applications. We further investigate with statistical data analysis and machine learning algorithms. Then we model the smart-phones as dynamic systems as in control systems to develop self-controllers for the participating systems. Then set of data preprocessing methods will be implemented. Then knowledge graph building process will take place. After this phase assembling orchestration will take place.

4.0.2 Future Work

Orchestrator is in essence the behaviour predictor of the participating devices with respect to time, location and as many as added contexts. The agent application installed in the smart-phones which report logs to the orchestrator will be facilitated with local validator and action triggers which is regularly updated by orchestrator according to the needs to avoid wasting orchestrator energy and data communication for well learnt case. In order to successfully deploy such orchestration service, we need to study and explore all its components defined in the previous section and their interdependencies in detail. Then we focus on the questions including 1) how to develop low energy consuming profiler? 2) how to reduce logs reporting thus by reduce data communication? 3) how to make orchestrator an epic predictor of device behaviours? 4) how to find optimal responsibilities of local agent by ensuring minimal computation ans resources? 5) Is it best fit for mass open source contribution? 6) What are the de-facto tools for over all implementation? We plan to implement and test the prototype iteratively. This work could be then extended or simplified to other type of IoT devices.

References

[1] Peter Flynn. *Formatting information. A beginner's introduction to typesetting with LaTeX*. 3rd edition, 2005. http://www.ctan.org/tex-archive/info/beginlatex/beginlatex-3.6.pdf.

[2] Patrik W. Daly Helmut Kopka. *A Guide to LaTeX 2ε*. Addison-Wesley Professional, 4th edition, 2003.

[3] Frank Mittelbach, Michel Goossens with Johannes Braams, David Carlisle, and Chris Rowley. *The LaTeX 2ε Companion*. Addison-Wesley Professional, 2nd edition, 2004.

[4] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. *The Not So Short Introduction to LaTeX 2ε*. 4.14th edition, 2004. http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf.

[5] Paul W. Abrahams with Karl Berry and Kathryn A. Hargreaves. *TeX for the Impatient*. Addison-Wesley Professional, 2003. http://www.tug.org/ftp/tex/impatient/book.pdf.