Demo: A Holistic Energy-Monitoring Framework for the IT Service Delivery Chain

Matthias Wichtlhuber*, Fabian Kaup*, Robert Reinecke*, Jeremias Blendin*, and David Hausheer*
* Peer-to-Peer Systems Engineering Lab, Technische Universität Darmstadt, Germany
Email: {mwichtlh|fkaup|reinecke|jblendin|hausheer}@ps.tu-darmstadt.de

***Abstract*—Energy efficiency of networking equipment has become an important factor for private consumers as well as business customers in the recent years. This trend is driven by two factors: the *proliferation of new mobile devices* as well as *energy prices as cost driver*. Battery capacity as well as dynamic distribution of workloads adapted to energy costs are an active field of research. This demo contributes to this discussion by presenting a framework allowing the assessment of energy costs along the content distribution chain.**

I. Introduction

The increased capabilities of modern mobile hardware have drastically increased the requirements for adequate battery capacity. Consequently, a recent study by uSwitch[1] revealed, that a long battery life is the most influential feature for the purchase decision of smart phone customers. Moreover, in the Information Technology industry sector, energy prices are a major influence on the operational costs of data centers and networking hardware. Thus, the world-wide placement of data centers is largely based on the climatic situation and energy cost in a certain country or region[2]. On top of that, mobile devices make up for higher backend utilization due to more complex routing and compressing strategies. In order to optimize the energy efficiency of mobile applications and cloud services for end users as well as service providers, knowledge on energy consumption is necessary, as shifting workloads in a non-optimal way may decrease energy efficiency of the system as a whole. In particular, a holistic view on the energy consumption of the service delivery chain including the service provider, the network provider, and the end user is required in order to decide on the distribution of workloads in the backend, a suitable network path and the communication patterns between mobile hardware and cloud services.

However, achieving such a holistic view is difficult. If traditional hardware based measurements are used, scalability as well as upfront investments are to be considered. Using this type of measurements requires a deployment of measurement hardware to a large number of servers, switches, and smart phones. While this might be costly but feasible in a data center or a backbone network, hardware based measurement reaches its limits when measuring mobile devices and collecting the information for a holistic view. Consequently, this work describes a platform for energy efficiency monitoring of the service delivery chain without the requirement of additional measurement hardware. Therefore, a model based approach is utilized: hardware measurements of devices under test are correlated to a number of parameters describing their workload. The resulting models are implemented in software and used to convert monitoring parameters into energy consumption values. In particular, the models of a smart phone [1], a home router [2], and a data center [3] were implemented for this demo described in II. Finally, the measurements are visualized for further analysis in III.

II. System Architecture

The monitoring framework consists of three main components as depicted in Figure 1. A group of light-weight System Utilization Monitors (SUMs) collects system parameter measurements. All monitors are based on a generic monitoring component encapsulating generic monitoring functionality. For each device type under test, an energy monitor is implemented taking the device's specific requirements into account. In particular, this demo features a monitor for virtual machines, a monitor for a home router, and a monitor for smart phones.

The collected measurement data is fed into the Energy Efficiency Monitoring Server (EEMS) using a REST interface. The EEMS is contacted by each SUM on startup and shutdown, which allows the EEMS to maintain a global view on the devices present in the network. Once the SUM has indicated the presence of the device, it starts sending measurements to the Power Controller, which converts the data to energy measurement estimates. For this, the measured system uti-

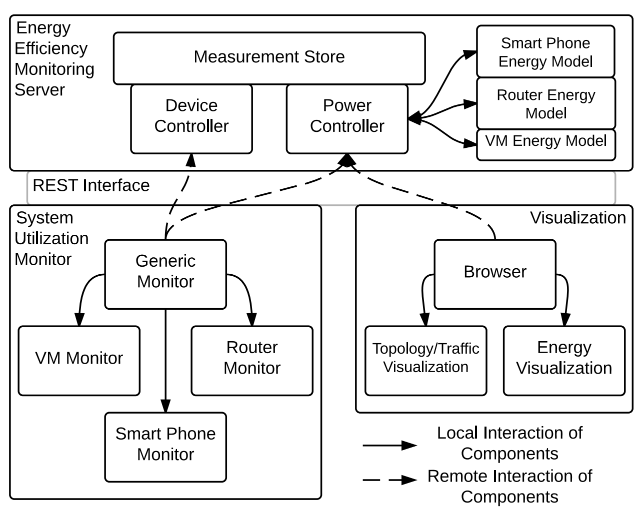

Fig. 1: Overview of demo architecture.

[1]http://www.statista.com/statistics/321432/smartphone-features-influencing-uk-consumer-purchase-decisions/, accessed: 2015-01-19

[2]http://www.datacenterknowledge.com/archives/2013/06/12/facebook-status-update-live-in-lulea-on-the-arctic-circle/, accessed: 2015-01-19

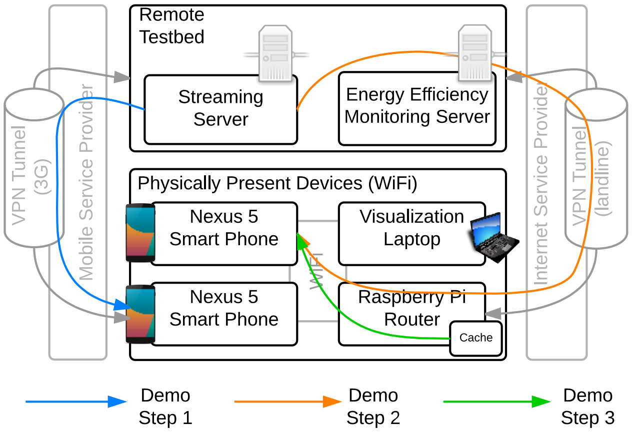

Fig. 2: Overview of demo scenario.

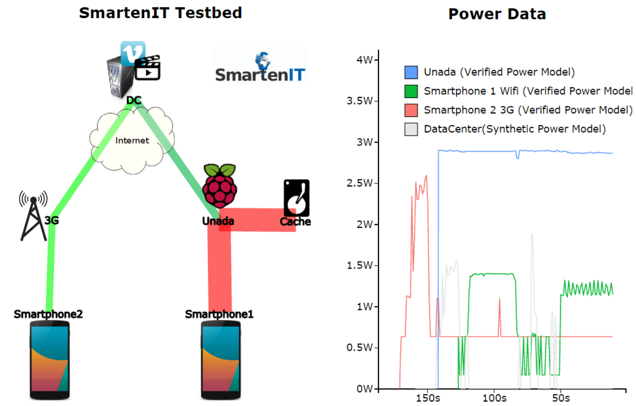

Fig. 3: Screenshot of visualization.

lization samples as collected by the SUM running on the Raspberry Pi are converted to power estimates as described in [2]. This power model includes the utilization of both interfaces and the CPU load. For the Nexus 5, only the network utilization is considered, as other parameters, like display brightness, are user and device dependent. The power model is generated similar to [1] and adjusted to exclude user and device dependent influences. The power consumption of the streaming server is approximated by a power model derived from [3], which is proportional to the generated traffic. The precision of the models is described in [1] and [2] respectively. Finally, the Power Controller stores the utilization samples and the energy estimates in the Measurement Store. The third component of the architecture is the Visualization component. This component runs in a browser and accesses the energy consumption estimates and system parameter measurements using the Power Controller. The data is visualized from the perspective of topology as well as energy consumption per device.

III. Demo Scenario

The central use case of the demo scenario is mobile video streaming. Scope of the use case is streaming via cellular access or Wi-Fi offloading by using a hotspot with caching functionality. As depicted in Figure 2, the demo scenario consists of six devices. Two Nexus 5 smart phones, a home router running on a Raspberry Pi platform, and a visualization laptop are physically present at the demo location. The physically present devices are interconnected using a local Wi-Fi access point provided by the Raspberry Pi router. Moreover, two additional virtual machines are running in a remote testbed, one hosting the EEMS, the other hosting a video streaming server. The remote testbed and the physically present devices are interconnected using two different VPN tunnels. One of the Nexus 5 smart phones sets up a tunnel to the testbed using 3G. The other devices can access the testbed using the Raspberry Pi router, which sets up the second VPN tunnel to the Remote Testbed using a landline Internet connection.

In order to demonstrate the capabilities of the monitoring framework, the demo is separated into three steps.

1) Video Streaming using 3G (reference case): A user streams a video from the Streaming Server using the 3G enabled Nexus 5 device. The video is delivered from the testbed using the 3G VPN tunnel connection. At the same time, the 3G traffic and energy consumption of the smart phone as well as the Streaming Server is monitored and visualized. The results serve as a reference case for comparison to the other steps of the demo.

2) Video Streaming using Wi-Fi: The user downloads a video using the second smart phone via Wi-Fi. While viewing the video, it is stored in the router's local cache. At the same time, the Wi-Fi traffic and energy consumption of the smart phone, the Raspberry Pi Router and the Streaming Server VM is monitored and visualized. As a result, the smart phone shows better energy efficiency than in the reference case, as the Wi-Fi interface is cheaper in terms of energy than the 3G interface.

3) Video Streaming using Wi-Fi and Caching: The user streams the same video as in case *2* using the same smart phone. This time, the video is served from the Raspberry Pi Router's local cache. Wi-Fi traffic and energy consumption of the smart phone and the Raspberry Pi Router is monitored and visualized side-by-side. The traffic visualization indicates that the video is served from the cache, i.e., the Streaming Server VM does not transmit data nor does it show any energy consumption. At the time, the energy measurements for smart phone and Raspberry Pi router stay roughly the same.

Figure 3 shows a screenshot of the resulting visualization of the latter case showing a lower overall energy consumption.

Acknowledgment

This work has been supported in parts by the European Union (FP7/#317846, SmartenIT).

References

[1] C. Gross, F. Kaup, D. Stingl, B. Richerzhagen, D. Hausheer, and R. Steinmetz, "EnerSim: An Energy Consumption Model for Large-Scale Overlay Simulators," *IEEE LCN*, 2013.

[2] F. Kaup, P. Gottschling, and D. Hausheer, "PowerPi: Measuring and Modeling the Power Consumption of the Raspberry Pi," *IEEE LCN*, 2014.

[3] "The Power of Wireless Cloud," Centre for Energy-Efficient Telecommunications, Tech. Rep., 2013.