Towards an Energy Lens on the Physical World with Mobile Phones

Jorge Ortiz
Computer Science Division
University of California, Berkeley
jortiz@cs.berkeley.edu

Abstract

Despite the recent impact of global warming and a steady rise in energy prices, per-capita energy consumption is rising. Part of the problem is about visibility. We simply do not have any good ways of seeing how we consume energy, and therefore, how to optimize and reduce it. Mobile smartphones present a unique opportunity to enable an energy view on the physical world. Smartphones provide a convenient bridge between the physical world, users, and computational infrastructure through its rich set of sensors, user interface, and ubiquitous connectivity. The camera, specifically, when combined with QR codes, gives us a portable scanner and convenient mechanism for tying these world together. In this paper, we describe our system and deployment experience for a mobile phone application the provides user-centric energy-view of the physical world. We describe the challenges, specifically dealing with mobility, and how we address them in a set of three separate applications: an energy auditing application, a device energy scanner, and a personal energy counter. We also introduce a technique called *dynamic aggregation* which allows us to seamlessly track the constituents of aggregated energy calculations, as they move from one location to another.

1 Introduction

The United States leads the world in per-capita energy consumption. Furthermore, our electricity use has consistently increased over the last 40 years [6]. With the specter of global warming and the increasing cost of energy, we must explore new ways for individuals to gain better visibility and insight about their energy consumption in order to optimize and reduce it. With the increasing penetration of embedded sensors in the environment and the the continued rise in the number of smartphone users, we see an opportunity to bridge the physical world to our computational infrastructure to provide a ‘energy lens’ on the physical world. This paper describes the design and implementation of a personalized, mobile energy visibility system. We discuss the architecture, experience, and challenges in a deployment we have done inside a campus building.

Smartphones serve as a natural point of intersection between the physical world, computation, and users. The camera serves as the input interface from the physical world, the screen functions as the interface to the user, and its connectivy serves to overlay virtual services on top of the physical capture to be presented to the user. In addition, since phones are highly personal items, they serve as a good proxy for information about the user that can personalize the views and services. Moreover, as mobile devices, views can be tailored to location and other contextual cues in the environment.

The use of mobile phones presents classical, fundamental challenges related to mobility. Moreover, objects in the physical world also move from place to place. This makes it even more difficult in trying to do fine-grained energy accounting. Not only must we deal with mobile users, but also mobile, energy-consuming objects in the environment. For example, to account for the total energy consumed on floor `X` at time `t` we must know all the energy-consumers on floor `X` at that time. Over time, this list changes, and these changes should be reflected in the total calculation at *any* point in time. Ultimately, physical-world state needs to be tracked by the application; it needs to know where things are and where people are so that it can answer queries about these things relative to where/who is asking it. Consistency management is non-trivial.

Our system uses QR codes to tag items in the physical world. Once tagged, there are three types of interaction: registration, binding, and scanning. During the registration phase a user scans the QR code and enters information about the item it is attached to. To bind items together a user scans multiple items. This creates creates virtual links between items that can be used to fetch or compute information about the item in relation to other items. For example, when a meter is bound to an item the data produced by the meter can be used a proxy for the item with respect to the measurements being taken by that meter. We used a network of wired and wireless meters and pushed their data into a cloud-based service that acts as a proxy for meters in the physical world. In addition, during the registration phase, metadata about each item is recorded that allows us to aggregate measurements in ways that are relevant to the user. For example, we can give the user visibility into the energy consumption of the room they are in, the item they scan, or the energy consumption of the items that belong to them.

Our system was deployed incrementally in a 7-story, 141,000 square-foot building. We tagged 351 items spread over 139 rooms throughout all floors. On this infrastructure we built an energy auditing application, a device energy viewer, and a personalized energy application. Based on the initial deployment experience we found that:

- QR codes are a convenient choice for tagging items because they are cheap, easily produced, and easily replaceable.
- Most smartphones are equipped with a camera and can download QR code scanning software for free, making it a effective, mobile scanner.
- Network connectivity is ubiquitous. Smartphones can connect to the internet through the cellular network or WiFi.

Although the infrastructure takes some time to set up at scale, the deployment can be done incrementally. Furthermore, once deployed, the infrastructure provides immediate value for all building occupants and sets a good foundation for building different types of applications. We describe some of the applications we built later in the paper.

Building applications that use the infrastructure reveal many challenges. Some common challenges include:

- Mobility. Maintaining user location (and context, more broadly) is difficult. However, tagged spaces throughout the building provide a check-in mechanism that lets us keep track a user's location over time. Still, users often forget to re-scan their location to their set spatial context. However, we learn a lot about context based on natural user action.
- Consistency maintenance between the physical world and virtual representation. Users forget to capture changes in item inter-relationships. For example, item A was once attached to meter M, but is now attached to meter *m*.
- Apportionment/accounting accuracy is difficult to control due to the challenges with mobility and consistency, but also due to metadata consistency over time (even when the mechanism that solve the first two problems, are correctly implemented). We introduce a feature called *dynamic aggregation* that keeps track of changes inputted by users to track the energy consuming constituents over time.

We discuss the various ways we have addressed some of these challenges in our deployment. Addressing these challenges is an on-going effort to reduce the interaction overhead between the user and her environment. An ideal solution would enable full, consistent visibility of the energy consumption of items in the environment as well as allow a user to effectively observe their own footprint, with minimal dependence on users maintain view consistency. Ultimately, we hope that detailed understanding of personal energy use will induce behavioral changes that reduce overall energy consumption.

2 Related Work

similar to [4, 8, 9].

3 System Architecture

Our architecture consists of various components, with the mobile phone serving as the centerpiece. Smart phones serves as a point of intersection between the user, her environment, and compontational infrastructure. Smarts phones are highly personalized, are carried by users everywhere, and with ubquitous, multi-modal forms of accessing a network, provide almost continuous access to services. However, although space and context are nebulous ideas and difficult to capture we can learn from user feedback. In order to truly bridge the physical world to the virtual world, we include the use of QR codes to tag *any item the user finds relevant to capture*. QR codes are easy to generate and inexpensive to produce and replace.

With energy tracking as the main goal, we used QR codes to tag items that consume energy. When an item is tagged, the user swipes the tag and enters information about the item. After this 'registration phase' is complete, any user can use their smart phone to learn about the item by swiping the QR code that is attached to it. In addition to arbitrary descriptive metadata, we distinguish between regular items and *meters*. Meters are items that produce continuous energy data. Meters are bound to items that are attached to them and serve as a proxy for the energy information about the device it is attached to. We describe how *binding* between items is done in section 4.1. Figure 1 shows our system architecture and each of the components.

3.1 Mobile phone

The mobile phone is at the center of our architecture. Its basic function is to swipe QR codes in the environment, do a lookup to the backend through the network, query for any relevant information to display to the user and display it. Interaction with the user is through this set of displays. For example, if a user wish to learn more about when their television was on, they swipe the QR code on the television. The phone offers various services that the television provides. In the current set of application we have written, you would get to read the descrip-

Figure 1. Architecture. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

tion, make, and model of the television. You might also learn what the rated power of the television, if that was included in the metadata capture. If the television has a meter attached and that meter is virtually and physically bound to it, on of our applications also allows you to view the power traces associated with television over various time intervals. The default is 24-hours.

3.2 QR Codes

A QR code is a two-dimensional barcode. You may encode up to almost 3000 bytes of data on them. QR code generators can be found on the internet [1, 2]. We encoded a lookup `URL` that our applications use to direct users to the appropriate web address and to identify the specific item that the QR code was attached to.

(a) Long QR Code. (b) Minimized QR Code.

Figure 2. The QR code on the left resolves to the same `URL` at the right one, after resolution and redirection is complete. The short label resolves to `http://tinyurl.com/6235eyw`. We used tinyUrl to reduce the QR code image complexity and scan time.

An interesting observation we made in our deployments is that complex QR codes are difficult to capture and resolve quickly, especially under poor lighting conditions. The more data you encode on the QR code, the more intricate the pattern that is generated, therefore we tried to minimize the size of the `URL` encoded on the QR code. The examples above show the difference between encoding a long `URL` to a short `URL` shows an example

	Average (sec)	Variance (sec)
Short,light	1.66	0.33
Short, dark	2.08	0.35
Long, light	2.26	0.71
Long, dark	2.82	0.50

Table 1. Shows the time to scan a long QR code versus a short QR code in light and dark conditions (loosely defined). Notice that short QR codes scan faster and with less variance that long ones.

QR code that we used to tag items in one of our deployments. Notice the difference between Figures 2(a) and 2(b). Since the first QR code is not only more complex, but also longer to scan than the second.

Table 3.2 shows the results of some simple scanning experiment between the two tags shown above. We scanned each QR code under light and dark lighting conditions, off the screen of my laptop. Each experiment was run 10 times and the table shows the statistical overview of the results. The absolute times are less important than the relative times. Clearly, scanning the simple QR code under well-lit conditions performed the best. In fact, the complex QR code under the same condition takes about 75% longer to scan. More importantly, however, is the variance. Notice that the variance with the simple QR code is much smaller and more stable under either condition. In our experience, **large variance in scan time is a major problem for complex QR codes**. That is why we decided to re-design them and push more information in the lookup processes, as network access was more reliable than the focus of the camera on various mobile devices. Tags are placed on all types of devices in all kinds of locations with varying degrees of lighting. Simple QR codes are vital for widespread use as a physically tagging mechanism.

3.3 Data infrastructure

The third piece of our architecture consists of a data collection and management components. The subcomponents of this part of the archictecture are set up in layered fashion. The context and data mangement layer is at the top, the meter-access layer is below that, and at the bottom is the metering infrastructure. The context layer sets up a structured metadata organization whose semantics are interpretted by each application. We describe the single structure that supports our applications. In addition, this layer unifies the metadata and the data, allowing the application to access all meters data that share similar properties (i.e. placement, type, owner, etc.). The meter data-access layer unifies the various access standards into a single interface. This allows us to add sensors and easily incorporate it into our context layer. Finally, we integrated various types of meter data. Specifically, we used several wireless power meters called ACmes [7] as well as various streams from the building management system directly.

3.3.1 StreamFS

We used StreamFS [3] as our data management layer. StreamFS offers various facilities to manage streaming sensor data and the associated metadata in a way that was useful for our the needs of our application. It organizes the metadata hierarchically and provides analytical facilities that are fully baked into the system, making it easier to create energy-analytics applications. In particular, StreamFS deals well with data aggregation where the number of constituents used to calculate the aggregate changes over time. In StreamFS, this is called *dynamic aggregation* and it is described in more detail in section 6.

Each object in physical space is represented by a node in StreamFS. Through StreamFS' RESTful/HTTP interface, we can fetch the any node through a path rooted at the location where StreamFS is hosted. For example, if we wish to access the power meter in room 400, we may access it by issuing an HTTP `GET` request to `http://server.streamfs.com/bldg/rm400/pow`. The request returns a `JSON` object with attribute-value pairs that give a description of the temperature sensor. It also returned that last received value from that sensor. StreamFS provides a query interface to fetch the time-series data collected from the sensor as well.

In addition, StreamFS support symbolic linking and this allows us to refer to nodes by multiple names. That same power meter can also be referred to through `/jortiz/meters/pow`; the meters that belong to user *jortiz*. More generally, StreamFS offers features that simplify namespace and data management. Semantically, the hierarchical node structure can be interpretted by the application. We describe the structure and interpretation in each application is section 5.

3.3.2 sMAP

sMAP [5] provides a uniform data access layer for sensor information. We can take any of the various sensor protocols and make it accessible through sMAP's HTTP, RESTful API. In addition, sMAP servers can live on any machine that is accessible through the web. This is convenient for our purposes, since it makes the raw sensor data stream accessible through any network. sMAP's data-forwarding facility is extensively to link the sensor data with the context layer. When a meter is being added to the context layer, a callback is installed on the sMAP server that hosts that meter's stream. As meter data is reported to the sMAP server, it is forwarded to the appropriate node `URL` in StreamFS, setting up the context associated with that meter automatically.

3.4 Metering layer

We used several types of meter data in our applications. Specifically we integrated the whole-building building power-draw feed, and a number of feeds from our wireless ACme power-meter deployment.

4 Setup and interaction

Figure 3 shows all the components of our architecture and the gestures used to interact with the system. To start we need to capture the objects and relationship between them. Specifically, we set up the hierarchical organization of objects with the building at the root, followed by *spaces* and *inventory* within the building, and data streams at the leaves. The spaces subtree is the hierarchical organization of floors, rooms, and areas on those floors. The inventory is a folder where each of the physical objects is placed. We used symbolic links to associate items in the inventory with the location they are in. In addition, we added a categorical subtree to ease aggregation with respect to the category of devices. For personalized accounting, users created their own folder that points to items that belong to them. Figure **??** shows a partial view of the hierarchy that supports our applications.

After the hierarchy is constructed, we place QR codes on objects in the world and register them. The registration process for locations includes the following steps:

1. Place QR code somewhere in the space.
2. Scan QR code.
3. Choose which space it represents.

The basic set of spaces that initially inputted by hand are the building and the floors. We placed QR code on the entrance of the building and the main door to enter the floor. Once each of these spaces is registered, the user can set their context by swipe the QR code tag. When rooms are added, the user first scans the tag for the floor they are on, then they place a tag on the room and they enter information about the room. For items, the process is the same. We scan the room we are in, place the tag on the object and enter information about object.

The initial location scan sets the context. For example, when you first enter a building, you scan the QR code associated with that building. The `URL` fetched from the QR code is first resolved. This is an example `URL` encoded in a QR code: `http://tinyurl.com/6235eyw`. When this is resolved, we get an empty response in the body, but we

Figure 3. Gestures. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

use the header to identify the QR code identifier that we associate with the item. The response header looks as follows:

```
HTTP/1.1 301 Moved Permanently
X-Powered-By: PHP/5.3.8
Set-Cookie: tinyUUID=ee81f56c2c15850975b7d175;
expires=Thu, 13-Dec-2012 04:00:18 GMT; path=/; domain=.tinyurl.com
Location: http://streamfs.cs.berkeley.edu/mobile/
mobile.php?qrc=4eb460a39fcd7
X-tiny: db 0.015100002288818
Content-type: text/html
Content-Length: 0
Connection: close
Date: Wed, 14 Dec 2011 04:00:18 GMT
Server: TinyURL/1.6
```

Figure 4. The header of the response from the `tinyUrl` **when resolving a QR code. The 'Location' attribute is used to extract the unique identifier for the object this QR code tags. It is also used to redirect users without the phone application to a meaningful web address for the object.**

As previously mentioned, we designed our QR code with the minimal amount of information to minimize scanning time. Therefore, we rely on the network to give us the rest of the information. This is a design choice. Network connectivity is more reliable than scanner quality, especially since we were dealing with a diverse set of phones and phones cameras of varying quality. The QR code contains a tinyURL, which resolves to a longer URL. The longers URL itself serve a dual purpose. It provides a web address for users to re-direc to and find information and various read-only services for the object. However, because the `URL` also contains a unqiue identifer *qrc*, it can be used to provide for sophisticated services and capabilities. An example is the ability to change the virtual structure of inter-relationship between this object and other objects. This is demonstrated in our energy auditing application discussed in detail in section 5.1. Once items are tagged, they can be added and removed by swiping the tag and pressing the button for what you want to do with the item. You also check into locations either explicit with a location-tag swipe or implicitly with an item swipe.

4.1 Inter-relationship capture

There is a special relationship between meters and items. Items are attached to meters, but more importantly, the data collected from the sensor *represents* the underlying dynamics of some physical measurement related to the item. A power meter attached to a television gives us the power profile of that television over some time period. Furthermore, if the meter is removed from the television and attached to another item, that change needs to be recorded, so that we do not attribute the power trace from the second item to the television. There are also items are that attached to each other that can affect how we aggregate feeds. For example, in our deployment, we sometimes connect meters to power-strips, which have multiple items attached to them. The meter serves as a proxy-aggregate for the attached the power strip that's attached the meter.

Figure 5. Attach and bind. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Both types of relationship are interpretted by how nodes are symbolically linked. If a meter is the child of an item, it is bound to that item and can be used as a proxy for measurements pertaining to that item. If an item is a child of a meter, it is attached to the meter, but the measure is not taking measurements pertaining to the item, instead if it taking measurement of the children of the item. Figure 15 highlights the two kinds of relationships interpretted by our applications. The one of the left shows a bind relationship while the one of the right shows and attach relationship. To bind or attach the gesture is the same. You first swipe the item the swipe the meter and press a button to either un/attach or un/bind.

5 Applications

In this section we discuss three applications we wrote that use the infrastructure we set up inside a building on campus. The building is a 7-story, 141,000 square-foot building. We tagged 351 items spread over 139 rooms throughout all floors. There are WiFi access point spread throughout the entire building, providing ubiquitous access to a network throughout. Cellular coverage is also available throughout most of the building as a secondary option for connecting to a network.

The first application we built is an energy auditing application. The goal was to get a sense for how energy was being used by the plug-loads distributed throughout the building. We collected both power rating information as well as live metering information. The second application, the energy-view item scanner, builds off the first, allowing users to swipe the QR code of an item and view the power-trace and power-rating recorded for that item. It is also used to view aggregates across space; you can scan a floor and see the aggregate data for that floor over time. The third is a personal energy accountant. This application allows users to virtually tag items that belong to them and view their personal energy consumption, aggregated across all the devices they own.

5.1 Energy Auditing

The energy auditing application was written to capture all of the plug-load devices in a building. Included it their capture is information about the plated power-draw of the device, if available, its make, model, and category. The capture consists of walking around the building with a set of QR codes, registering each of the rooms and items within them. For each room you enter the name of the room. The floor is indicated explicitly through a swipe of the buidling floor tag. Once inside the room each electrical device is tagged and registered. To register the item the QR code needs to be added, followed by a small form that is filled out by the auditor. The form includes information for the name of the device and the power rating or current draw.

Figure 6 shows two screenshot of the auditing interface. The first page is a list of options and the second is the registration page. Once the page is filled out, the user scans the QR code once again and the item is added to the inventory. In addition, the QR code tag identifier is added to the 'qrc' directory and symbolically linked to the newly added item in the inventory directory. Finally, a symbolic link is create in the directory for the room that also points to the item just added to the inventory.

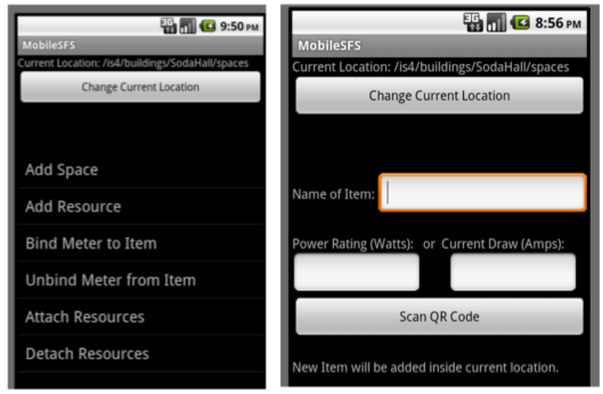

Figure 6.

Auto-classification was done using the same of the item as well. In addition to the spaces, qrc, and inventory directories there is also a taxonomony directory. Items with a given prefix were classified as being items in a particular directory of the taxonomy. A symbolic link was also created from the corresponding node in the taxonomy, that most closely matches the item, to the item node.

5.1.1 Results

This application was focused on capturing the physical, energy-consuming objects, describing them in various way and coupling that information with live streaming power data. Figure 7 shows the breakdown of the types of devices that were recorded by the energy auditing application. Since this data was collected from a office building, most of the device were actually LCD screens.

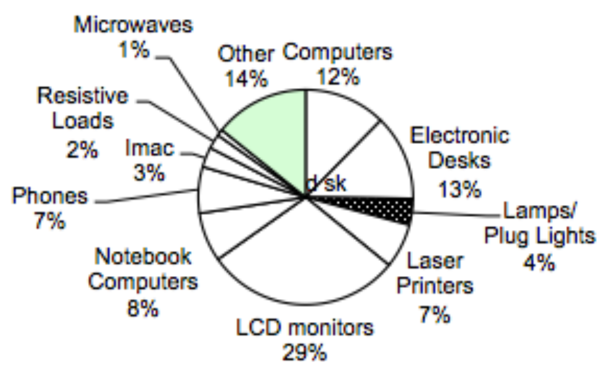

Figure 7. The percentage breakdown of the devices that were captured.

Although most of the items are static, a good number of them are moved by their owners pretty often. For example, 8% of the device were classified as notebooks. When the owner leaves the room or building, they usually take their notebook with them. This kind of information should be recorded using the auditing application. By simply swiping QR code for the notebook and clicking the 'leave' button, the item is kept in the inventory for the building, but the symbolic link from the room to the item is deleted.

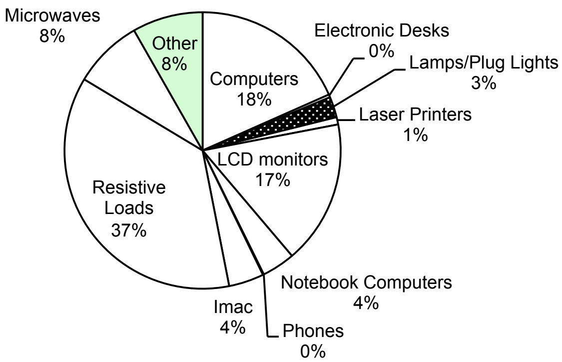

Figure 8. Device power draw in Watts.

Figure 8 shows the power-draw distribution. Interestingly, although the category of devices that fall under 'resistive loads' only make up 2% of the items recorded, they account for 37% of the total power-draw. However, note that this is calculated with respect to the power-draw figure placed on the device itself. It is not based on measured power-draw. This calculation was made using the properties that were inserted in StreamFS. Each of the item nodes were recorded as having a certain power-draw and that was used to calculate the total the percentage of total according to each category.

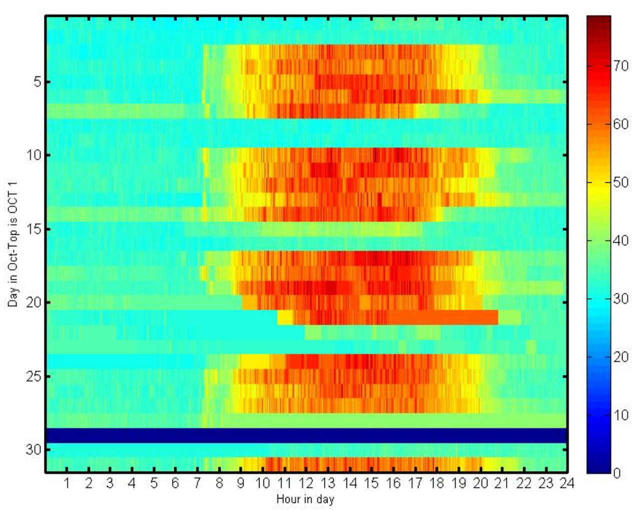

Figure 9. Power heatmap generated from the data obtained from meters attached to plug-load devices throughout the building. Red zones are locations where the highest amount of power is being consumed.

Figure 9 was generated using actual live, plugload data throughout the day during the entire month of October. The x-axis shows the hour of the day, while the y-axis indicates the day in october. This data was aggregated over hourly period by summing all the plug-load data collected over that hour. Notice the busiest times between 10am and about 6pm. There's also a clear view of the weekends.

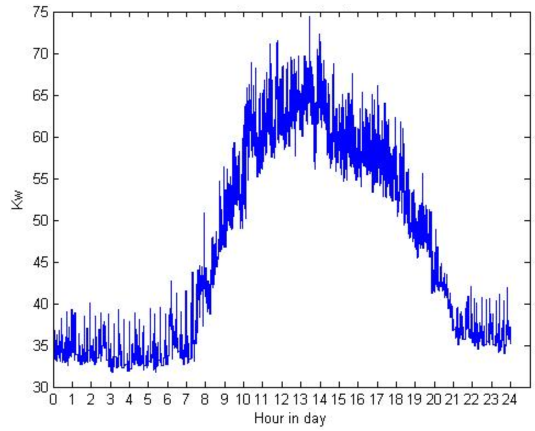

Figure 10. Whole building power draw on October 12th.

Figure 10 shows the whole building power-draw feed. This meter was obtained through the building management system and added to the audit. Notice how the peak and low times correspond to the patterns observed in the monthly plug-load heat map.

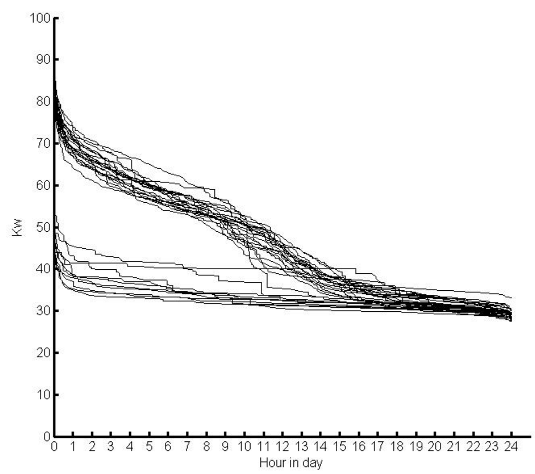

Figure 11. Load duration curves.

Figure 11 shows the load duration curve for each day in the month of October. The load duration curve shows the number of hours in a 24-hour day that the load was at or above the level indicated on the y-axis.

5.1.2 Issues

Although many of the calculations are static and in large aggregates there were some fundamental challenges that we encountered. The first challenge is maintaining *consistency* between the physical world and the virtual representation of it. About 16% of the plug-load items we registered can be moved between locations by they owner with relative frequency. 7% (laptops) move quite often. Maintaing of accurate view of what items are where is a non-trivial with the right mechanisms in place. Our approach in this case is to depend on the ubiquity of smartphones and participants. If a person moves an item from one location to another, they can swipe the item out of the current location and swipe the item back in at the new location. In addition, we take advantage of natural usage patterns. People tend to forget to swipe items out of their current location, so we leverage the second swipe (swipe-in) to imply the first operation (swipe-out). If the item was connected to a meter, we unbind the item from the old meter before swiping the item out.

We are currently working on adjusting the timestamp based on this action as well. From the trace of the meter we can see when the item was unplugged (or turned off). The sudden drop in power reading could be used to mark the point of disconnection. However, this only gets us part of the solution. If the person forgets multiple actions, it become more difficult to determine what has occurred. For example, if a lamp is plugged into a meter in room 1 and they unplug it from the meter, plug another device into it, and move the lamp to another location, the unplug time is not clear. If the new device draws power, we cannot tell the difference between the trace generated from the new device versus the old device. This is a non-intrusive load-trace classification problem and beyond the scope of the current project.

5.2 Item Scanner Application

The next two application were built on top of the deployed infrastructure from the energy audit. The item scanner application shows a power trace of an item. If the item is a space, it shows the aggregate consumption of the space over a 24-hour period. Figure 12 shows a screenshot of a trace for a room in on of the spaces we monitored.

The main thrust of this application revolves around aggregating feeds with respect to the spatial hierarchy that was constructed for the energy audit. A person can scan any item from the building and floor down to the room and individual device.

5.2.1 Issues

In addition to the consistency challenges from the energy audit, apportionment is non-trivial. Even with an accurate view of where items have been moved, tracking the consituents and calculating aggregate is challenging in real-time. Since meter clocks are not syncrhonized, the data must be cleaned before an aggregate can even be computed. We address this problem with dynamic aggregation and discuss the details in section 6.

5.3 Personalized Energy Tracking

A user identifies themselves with a username and password. This create a folder in StreamFS with their username. As they walk around the building, they can tag items that belong to them by swipe them and clicking ‘my item’ in the mobile application. The application records the item and its location when this is done. A location folder is created in the user’s directory along with a symbolic link to the item that they tagged. This allows us to aggregate both the totals by location and totals by user. Figure 13 shows the interface for tagging a device and the aggregate for the owner of the items.

Figure 12. Item scanner screenshot. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry’s standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Figure 13. Personal energy screenshot. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry’s standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

5.4 Issues

In this application, the main challenge was in localization of a particular user. In order to form aggregates for the total c onsumed by the user in a particular sapce, we want to present the user with an aggregate of the items they know in that space. To do this automatically one might require indoor localization using the WiFi infrastructure. However, we leverage the tags infrastructure and user input to identify who they are and where they are. With a simple 'login' and swipe, they get personlized, location-specific, aggregate energy consumption data.

- Mobile objects: objects move and are placed in different locations
- Changing meters: meters are moved or items are disconnected and re-connected to other meters
- Aggregation that track constituents over time: just a description, talk more about it in next section

6 Dynamic Aggregation

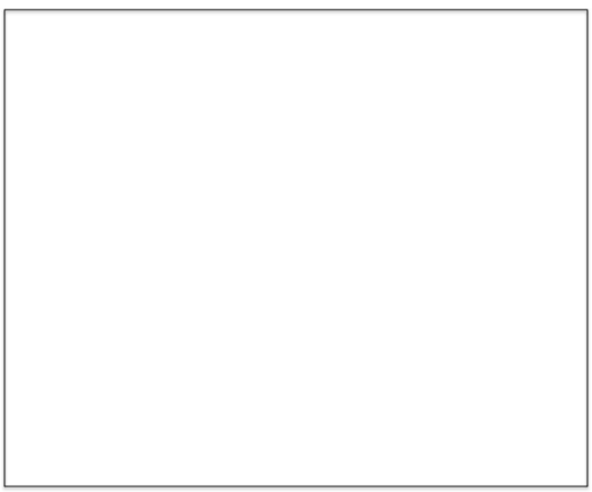

Figure 14. Aggregation tree. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

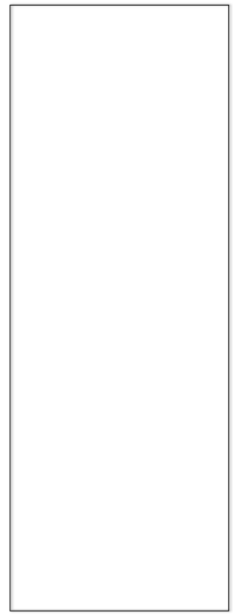

Figure 15. Stretch and fill operations. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

7 Lessons learned

8 Conclusion

9 References

[1] `http://qrcode.kaywa.com/`.

[2] `http://www.qrstuff.com/`.

[3] `http://streamfs.cs.berkeley.edu/`.

[4] D. Caswell and P. Debaty. Creating web representations for places. In *Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing*, HUC '00, pages 114–126, London, UK, 2000. Springer-Verlag.

[5] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler. smap: a simple measurement and actuation profile for physical information. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 197–210, New York, NY, USA, 2010. ACM.

[6] *Energy Balances of OECD Countries*. International Energy Agency, `http://www.oecd-ilibrary.org/energy/energy-balances-of-oecd-countries-2011_energy_bal_oecd-2011-en`, 2011.

[7] X. Jiang, M. V. Ly, J. Taneja, P. Dutta, and D. Culler. Experiences with a high-fidelity wireless building energy auditing network.

[8] T. Kindberg and J. Barton. A web-based nomadic computing system, 2000.

[9] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10:91–102, January 1992.