Exhausting Battery Statistics

Understanding the energy demands on mobile handsets

Narseo Vallina-Rodriguez†, Pan Hui⋄, Jon Crowcroft†, Andrew Rice†

† University of Cambridge
15 JJ Thomson Avenue
Cambridge, UK
name.surname@cl.cam.ac.uk

⋄ Deutsche Telekom Laboratories
Ernst-Reuter-Platz 7
Berlin, Germany
name.surname@telekom.de

ABSTRACT

Despite the advances in battery technologies, mobile phones still suffer from severe energy limitations. Modern handsets are rich devices that can support multitasking thanks to their high processing power and provide a wide range of resources such as sensors and network interfaces with different energy demands. There have been multiple attempts to characterise those energy demands; both to save or to allocate energy to the applications on the handset. However, there is still little understanding on how the interdependencies between resources (interdependencies caused by the applications and users' behaviour) affect the battery life. In this paper, we demonstrate the necessity of considering all those dynamics in order to characterise the energy demands of the system accurately. These results indicate that simple algorithmic and rule-based scheduling techniques [7] are not the most appropriate way of managing the resources since their usage can be affected by contextual factors, making necessary to find customised solutions that consider each user's behaviour and handset features.

Categories and Subject Descriptors

C.4 [Performance of systems] Measurement techniques

General Terms

Measurement, human factors

Keywords

Smartphone usage, user behaviour, resources demand

1. INTRODUCTION

Simultaneous use of the diverse hardware systems embedded in a modern smart phone would limit many handsets to just a few hours of operation. In practice a phone will attempt to mitigate this problem and extend its lifetime by making selective use of the available resources. This is most often implemented through the use of standby power states, automatic control of the screen backlight, and actively switching particular subsystems (such as networking technologies) on or off as demand dictates. These techniques are demand driven and so it is quite possible for a power-hungry application to drastically shorten the operating time of the handset. Power-aware operating systems such as Cinder [16] attempt to alleviate this problem by enforcing energy allocations made to particular processes. However, the complex and rapidly evolving way in which we interact with our handsets makes this allocation a difficult and dynamic problem.

The Google Nexus handset is a pertinant example. This device contains a 1GHz ARM CPU with additional hardware support for various network technologies (e.g. GSM, UMTS, HSDPA, HSUPA, Wifi and Bluetooth), embedded GPS and A/V accelleration. Not only does this comprise a complex platform but there are often many different opportunities for achieving some particular goal each of which provides a different tradeoff in power consumption and performance. This makes previous energy models and resources managers designed for laptops [20] and data centers [4] inapplicable. Applications such as Google Latitude [11] create further complexity by generating correlated demand across many disparate subsystems of the phone.

In order to better understand the resource management challenges posed by these devices we ran a preliminary study collecting data on handset usage from a small set of volunteers. In this paper, we use our study to argue that system workload, resource utilisation and energy demands are diverse and dynamic both in time and space, are highly affected by contextual information, and vary significantly for individual users' patterns of usage. The ramifications of this are that the largely static, uncorrelated allocation systems used in systems such as ECOSystem [20] and Cinder [16] are likely to be very difficult to use in practice. We highlight strict usage routines evidenced by some users as they interact with their handsets at specific places and times. For these users it is possible that this kind of contextual information will prove a useful input to any energy allocation algorithm.

Our particular interest is in the construction of a Social Operating System which not only uses the hardware within a device to efficiently achieve some goal but which also shares this functionality between devices. This study is our first work towards discovering the plausibility of such a system which will depend on the manner in which smart phone handsets are used, the demands of their applications and the energy costs thereof.

2. DATASET DESCRIPTION

The data collected in our experiment consists of time series values collected by tracking the mobile usage of 18 *Android OS* users for a period of 2 weeks in February 2010. Most of the participants used their personal handsets; however, three *Android* phones were given to volunteers. No constraints or limitations were imposed

MobiHeld 2010, August 30, 2010, New Delhi, India.

Dataset Description	
OS Info	Total CPU Load OS and Process CPU Load Process Running Free/Cached Memory
Battery	B. Level B. Temperature B. Current B. Voltage Charging Status
Network	Airplane Mode Network Type Data Network State Data Network Activity Roaming Calling State WiFi State Bluetooth State Network Traffic CID/LAC
Others	GPS Status Screen Activity USB Connection

Table 1: Information collected for Android handsets

Power State	Telephony	Power (W)	Power with daemon running (W)
Standby	Airplane	0.0199	0.0714
Screen On	Airplane	0.3416	0.3857
Standby	Cell	0.0322	0.1157
Screen On	Cell	0.3739	0.4099

Table 2: Measured energy consumption for an HTC G1 with and without Resources Tracker running in the background with different screen and telephony modes

on the subjects and they were encouraged to interact normally with their phones. In total, we collected a total of 275 days of active mobile usage and a further 70 days of inactivity for which the phones were switched off, mainly at night whilst charging.

The information was collected using Resources Tracker, a background process running on each handset which sampled the status of more than 20 variables highlighted in Table 1 every 10 seconds. We used an alarm callback to suspend the process between samples in order to minimise the impact on battery life and system performance. Most of the information was obtained by using the Android public APIs[1] and the information available in the */proc/* filesystem. The information was stored locally so no network usage was required and the location information was obtained using Cell ID information, a passive method that does not require turning on any energy intense sensor such as GPS.

As it is shown in Table 2, the additional power consumption caused by the background process is not inconsequential and a number of users reported that they noticed a shortened battery lifetime. In future we hope to investigate whether it is possible to reduce the overhead of logging without impairing data fidelity by varying the sampling rate perhaps based on state change events from the handset.

3. USAGE PATTERNS

Due to the multivariable nature of the problem, we were first interested in discovering dependencies between different handset resources in order to reduce the analysis complexity. This section explains how we used two well known multivariable analysis techniques [17] Principal Component Analysis (PCA) and Factor Analysis (FA) for this purpose.

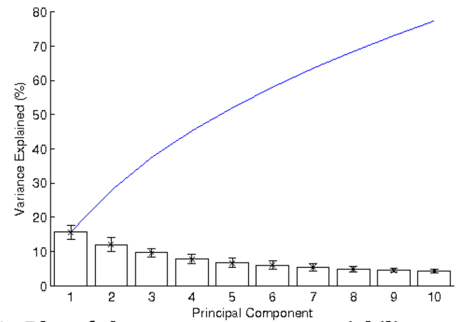

Figure 1: Plot of the average percent variability explained by each principal component considering all the *Android* users.

PCA is a mathematical tool commonly used to transform a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for as much the variability of the data as possible, and each subsequent component accounts for as much of the remaining variability in turn. FA is a complementary statistical method used to describe variability among observed variables in terms of fewer unobserved variables called factors. The observed variables are modeled as a linear combination of all the factors, plus *error* terms. FA estimates how much of the variability is caused by common factors whereas PCA estimates which factors can account for as much of the variability as possible.

After running PCA for each user, it is clear that it is not easy to reduce the number of variables and the complexity in the system. Figure 1 shows the percentage of the variance that each one of the first ten principal components can explain on average for all the users. It is necessary to consider up to 10 variables to explain almost 75% of the system variability. Furthermore, those 10 principal components also differ between users. Figure 2 shows the coefficients obtained from PCA and FA for 2 particular users. The principal and second component are represented in this biplot by the horizontal and vertical axis respectively while each of the variables coefficient is represented in this plot by a vector which its direction and length indicate how this variable contributes to each one of the two principal components. The significant factors are markedly different for each.

Although there are no systematically useful simplifying factors across our dataset we can use these results to distinguish between participants with different patterns of usage. We can identify, for exampe, which users did not take advantage of particular resources and services. Some users never took advantage of power-saving features like airplane mode and others never were roaming. PCA can detect this diversity of usage between users. Moreover, the PCA analysis revealed that GPS and voice calls do not account for much of the variance in the dataset. As a result, they are not responsible for as much of the total energy consumption in the system as use of the 3G, WiFi and the screen backlight due to their low usage. Moreover, FA also reveals that some sets of variables are highly correlated to each other (e.g. *battery capacity*, *charging mode* and *battery voltage*) independently of the users, thus it is possible to slightly reduce the number of variables to study. Nevertheless, the results obtained by these techniques lead us to argue that one should view usage from a user-centered perspective, rather than attempting to cluster the general behaviour of the users as in previous studies [14], [3].

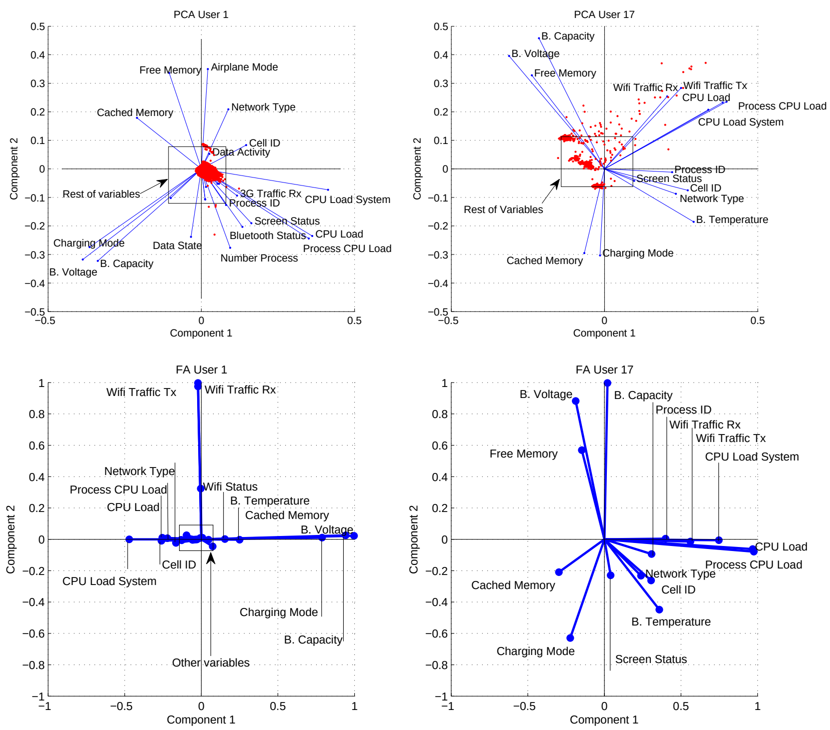

Figure 2: Biplot showing the principal and second principal component coefficients (Top) and the Factor Analysis coefficients (Down) for two *Android* users: *U1* and *U17*. Those techniques clearly reveals the notable differences between the users interaction with their handsets

4. THE IMPORTANCE OF CONTEXT

It is well known from previous studies that user mobility can be highly predictable both in cellular networks [18] and in WLANs [19]. Users generally remain subscribed to a small set of base stations and the majority of any interaction with their resources or applications takes place there. This makes it possible to detect locations where the energy demands are higher. Figure 3 shows the percentage of time that the users were subscribed to their 3 most common basestations during the experiment[1] and the percentage of time that the OS did not report any base station information[2]. During the whole experimental period, an average user spent more than 50% of their time subscribed to their top 3 cells and conversely for 25% of the time no cell information was reported.

Energy demand and resource availability depended enormously on each participant's pattern of usage both in terms of which applications they ran and when and where they were doing so. This interaction can be very variable and dynamic both in time and space. For instance, some users demonstrated heavy 3G data usage whilst others spent a lot of time calling. Nevertheless, an average user from our study sent between 5 to 10 MB of data per day, called 5 to 10 minutes a day and had between 30 to 90 minutes the screen active per day.

Spatial context affects how users interact with their handsets. For instance, as an extreme case, a roaming user will reduce or eliminate their 3G traffic for financial reasons. Figure 4 shows three scatterplots of the average percentage of daily usage of the 3G interface, telephony and the screen while the users are subscribed to their three most popular cells. We can infer that users *U1*, *U5*, *U8*, *U9*, *U14* and *U18* have a strong routine due to their low variance and are quite likely to interact in those locations, presenting a highly predictable usage. On the other hand, further research is necessary to understand the remaining users to identify whether

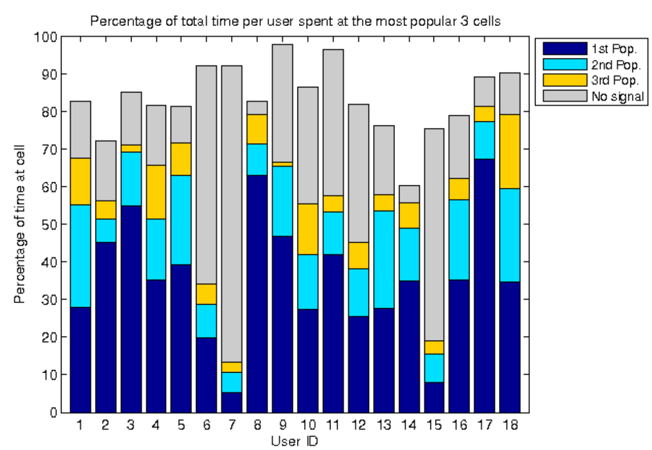

Figure 3: Experiment subjects' cells subscription during the length of the experiment. Note that *No Signal* does not strictly imply that the handset is not under network coverage: the OS could have failed at the time of providing this information.

[1] This graph only includes the time that the phones were active.

[2] CID, MAC, MMC and MNC parameters. Note that *not reporting* cell information can be caused by errors when polling the OS and also by the periods of time without signal. Nevertheless, in case not having access to the cellular network, confirms that energy consumption is highly influenced by environmental factors like geomorphology and demography

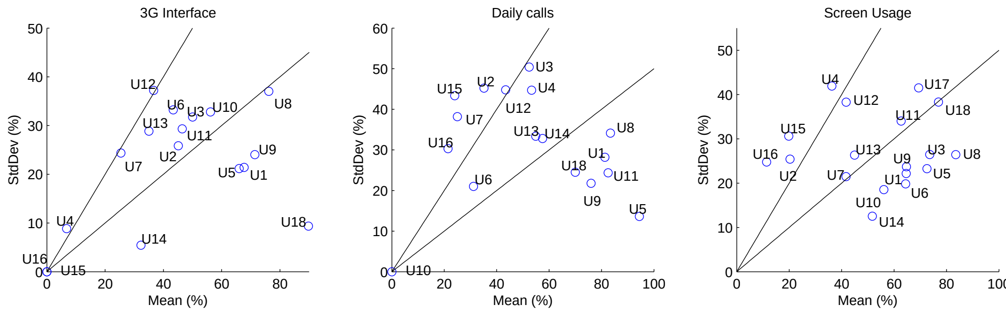

Figure 4: User classification by their percentage of the usage/interaction with the 3G interface, phone and screen while subscribed at the most popular cells. That information can be used to identify the places where the energy consumption will be higher.

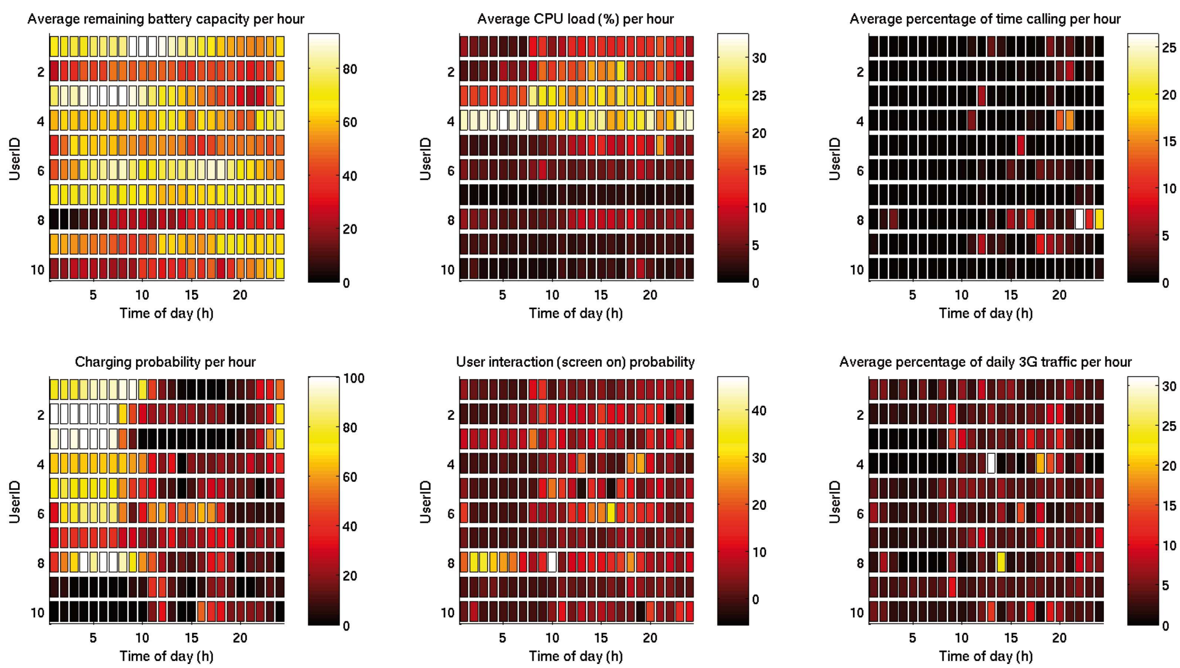

Figure 5: Average usage and availability of different mobile resources for users *U1* to *U10* per hour of day. As with Figure 4, that information can be used to identify peaks of usage on those resources and the time when the energy resources can be recovered.

they are interacting with their handsets whilst commuting, in defined non-popular places or simply in random locations.

Most resources in the handset can be recovered and re-allocated once used by a process. However, that is not the case for energy which is only recovered by charging the handset. Figure 5 plots the average usage and availability of different mobile resources such as battery, telephony, network, screen and CPU for users *U1* to *U10* per hour of day. The results shown in Figure 5 reveal that the battery usage and charging opportunities are well defined for some individuals. This makes it easy to estimate when energy will be consumed, how much energy will be available and when it will be recovered. For instance, *U3*, regularly has minimum battery capacity late in the evening just before, with high-probability, she starts charging the handset. However, other users do not present such a defined pattern (subjects *U2* and *U5*) and yet others present a much burstier pattern for other resources. We highlight this in Figure 6 that plots the *correlograms* or *autocorrelation plots* of the battery capacity and the CPU load for 3 users for a 7 days lag. A correlogram is a plot of the sample autocorrelations versus the time lags that helps to identify randomness and periodicities in a dataset. The correlogram clearly reveals that *U3* presents a clear charging periodicity of 24 hours approximately while *U8* does not have such a marked routine. However, those results highly depend on the resource analysed since, as we can also observe, the CPU load is not periodic at all indicating that CPU load might be more difficult to predict than battery capacity.

5. ALLOCATING RESOURCES

Current mobile operating systems are multitasking and more than 250 different application processes and 68 system processes were reported amongst our Android users. This makes for an enormous set of of possible combinations of running applications. In this environment energy-allocating operating systems need to make efficient and autonomous allocation decisions whilst minimizing reliance on user input and inaccurate information as battery discharging rate. The use of off-line information as in Cinder [16] and

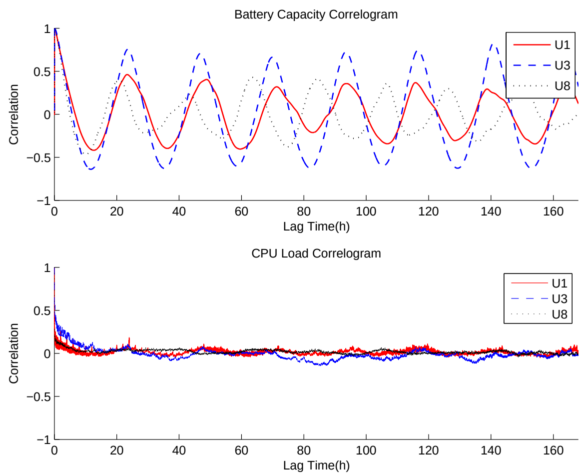

Figure 6: Correlogram for the CPU load and the battery capacity for users *U1*, *U3* and *U8* during a period of 7 days. The battery capacity correlogram shows a clear pattern and a periodicity on the energy consumption and recharging cycle every 24 hours approximately while the CPU load is highly random.

ECOSystem[20] is unlikely to succeed. Instead it would be interesting to see what can be learnt from application runtime behaviour.

Table 3 shows the *resources fingerprint* of the most intense applications in terms of CPU load, screen usage, GPS and network usage. As we can see, some applications are extremely intense on the usage of their resources. For example, Google Goggles is an image based search tool with process name *com.google.android.apps.unveil*. We can see it requires considerable processing power and network bandwidth. However, these statistics cannot be generalised for all the users since CPU intense applications are not necessarily intense in other dimensions such as network traffic and depend on the users' interaction with this application. This kind of statistics must be monitored at runtime by the OS in each handset to optimise the battery capacity of the device and if an application is not meant to be running, it must be killed. These decisions are difficult because there are many desirable application background processes such as email(i.e reducing checking) applications or streaming services like *Lastfm* or *Spotify;*. Nevertheless, the results provided in the previous sections clearly demonstrate that contextual information helps to identify the peaks of usage both in space and time, being possible to identify strong patterns of usage in some users. This information can be used by schedulers to efficiently allocate energy to the applications by forecasting the future demands.

6. RELATED WORK

Most of the studies conducted to understand the use and energy demands of mobile handsets can be divided in two groups. The first type focus on measuring and modelling the impact of wireless network interfaces on the battery life [15],[13] and providing techniques to extend the battery life of the handsets [2]. Prior work has also studied the impact of different energy saving techniques in 3G networks using analytical models [12],[6]. The second type of study addresses how users interact with batteries both in laptops and mobile handsets from a context-aware perspective [3]. Similarly, the authors of [14], propose a context-aware battery manager that predicts when the next charging opportunity will be and warns

CPU Load (%)		
Process Name	Avg	Max
com.skype.android.lite	86.9	95
com.glu.android.bonsai	73.4	82
com.markspace.missingsync	65.5	94

Active while Screen ON	
Process Name	% of time
com.htc.soundrecorder	99.8
dk.logisoft.aircontrol	99.7
au.com.phil.mine	99.7

GPS ON probability while running	
Process Name	% of time
com.cooliris.media	7.14
com.aws.android	6.01
com.google.android.apps.finance:remote	1.28

Downstream traffic (kbps)		
Process Name	Avg	Peak
cmupdaterapp.ui	172.2	767.2
com.google.android.voicesearch	144.3	249.6
com.google.android.apps.unveil	39.1	184.1

Upstream traffic (kbps)		
Process Name	Avg	Peak
com.google.android.apps.unveil	51.1	101.7
com.htc.album	27.2	214.1
comupdaterapp.ui	15.9	24.5

Table 3: Top 3 Applications by average resource usage

the user when it detects that the phone battery will be exhausted before this can happen. However, the complex interdependencies between resources limit the applicability of these studies to modern handsets. Location sensing is an example of the potential for using different sensors to provide the same type of information. In [21], the authors suggest a middleware system that aims to optimise the energy consumption by using measurements from inertial sensors to minimize the use of the GPS sensor.

There are many bibliographic references regarding resource allocation and energy-aware operating systems. Early work in operating systems such as Beos [5] could handle multimedia applications that were not very computation or I/O intense but had other characteristics in other dimensions. However, Beos was scheduling for a single goal: increasing the throughput between resources such as screen, disk and network interfaces. Flinn *et al.* also introduced Odyssey [8], an energy-aware OS and a fine-grained energy usage profiler by application as in the same way that CPU profilers such as *prof* map CPU cycles to specific processes [9].

Cinder [16] is an energy-aware OS for mobile phones focusing on allocating energy resources to applications. Cinder makes it possible to subdivide the energy share of an application among its constituent subtasks and allows users to set up their own policies. However, it is not clear how the energy required by each application or the energy demands of the multiple resources is derived in the first instance. Cinder takes inspiration from ECOSystem [20] and Quanto [10]. Ecosystem is an OS that supports energy as a first-class operating system resource and combines user preferences with resources monitoring to extend battery life by limiting the average battery discharge rate, sharing energy proportionally among different tasks whilst Quanto is an energy profiler for embedded network devices, mainly focused on sensors.

7. CONCLUSIONS

Energy is commonly reported as the primary target for optimisation in mobile handsets. Many researchers make energy a central element of the operating system, and their studies model the energy demands of the different hardware features. However, these studies rarely consider the dynamics and interdependencies among resources caused by applications and users' behaviour.

In this paper, we demonstrate the need to consider all these dynamics to characterise the energy demands of the system accurately. Energy allocation must be customised to each user and handset, and cannot be based on off-line information model. We find preliminary results indicating that users interactions with the mobile handsets vary enormously, but that it is possible to identify where and when some resources may be in high demand. To conclude, full contextual information helps make more efficient energy use.

These results demonstrate that algorithmic resource control is not efficient because of the number of diverse factors that determine the resources demand. We are now working on better prediction of resource demand, and applications' behaviour, trying to investigate more the different relationships between the resources and their nature. This is a necessary step before researching on non computational-intense techniques to predict the energy consumption and context-aware resources management. One approach may be to use machine learning techniques, to build a task killer and resource manager that allocates energy to applications based on their *fingerprint*, using periods of time where the handset has plenty of energy and computational power to perform those tasks.

Resources might be replicated and requested simultaneously in a specific place and time. Forecasting techniques are also a first step in order to understand how to allocate resources between users in an opportunistic fashion as our current work on *ErdOS*, a energy-aware social operating system, aims to do. Thanks to *ErdOS*, handsets will be able to share their local resources (e.g. network interfaces, sensors, CPU and storage) with nearby devices using low-power connectivities in order to increase the handsets usability and minimise the energy consumption. As a consequence, handsets need to act proactively to advertise which resources they can offer to their vicinity and also to forecast their future demands. The applications of a Social Operating System are multiple: handsets can collaborate to share a computation, they can co-ordinate for a particular purpose like multipoint recording of an event or sensing the environment, or even simpler, a single handset can be used as a server (e.g. location service) to other devices in the same locale.

Acknowledgments

We are grateful to the volunteers that lost almost all their privacy during two weeks. We wish also to thank Microsoft Research Cambridge and Google Inc. for lending us some handsets and Christos Efstratiou, Cigdem Sengul, Richard Mortier, Damien Fay, Kiran Rachury and James Scott for their insightful comments and help.

8. REFERENCES

[1] AndroidReference. `http://developer.android.com/reference`.

[2] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proc. of the 9th ACM SIGCOMM IMC*, 2009.

[3] N. Banerjee, A. Rahmati, M. Corner, S. Rollins, and L. Zhong. Users and batteries: Interactions and adaptive energy management in mobile systems. In *Proc. of UbiComp*, 2007.

[4] G. Banga, P. Druschel, and J. C. Mogul. Resource containers: A new facility for resource management in server systems. In *Proc. of the 3rd ACM OSDI*, 1999.

[5] BeOS. `en.wikipedia.org/wiki/beos`.

[6] M. Chuah, W. Luo, and X. Zhang. Impacts of inactivity timer values on UMTS system capacity. In *Proc. of WCNC*, 2002.

[7] C. Efstratiou, K. Cheverst, N. Davies, and A. Friday. An architecture for the effective support of adaptive context-aware applications. In *Mobile Data Management*. Springer, January 2001.

[8] J. Flinn and M. Satyanarayanan. Energy-aware adaptation for mobile applications. *Proc. ACM SIGOPS*, 1999.

[9] J. Flinn and M. Satyanarayanan. Powerscope: A tool for profiling the energy usage of mobile applications. In *Proc. of the 2nd IEEE WMCSA*, 1999.

[10] R. Fonseca, P. Dutta, P. Levis, and I. Stoica. Quanto: Tracking energy in networked embedded systems. In *Proc. OSDI*. USENIX, 2008.

[11] GoogleLatitude. `www.google.com/latitude`.

[12] C.-C. Lee, J.-H. Yeh, and J.-C. Chen. Impact of inactivity timer on energy consumption in WCDMA and cdma2000. In *Proc. of WTS*, May 2004.

[13] A. Rahmati and L. Zhong. Context-for-wireless: context-sensitive energy-efficient wireless data transfer. In *Proc. MobiSys*, 2007.

[14] N. Ravi, J. Scott, L. Han, and L. Iftode. Context-aware battery management for mobile phones. In *Proc. IEEE PERCOM*, 2008.

[15] A. Rice and S. Hay. Decomposing power measurements for mobile devices. *Proc. IEEE PERCOM*, 2010.

[16] S. M. Rumble, R. Stutsman, P. Levis, D. Mazières, and N. Zeldovich. Apprehending joule thieves with cinder. In *Proc. MobiHeld*, 2009.

[17] G. A. F. Seber. Multivariate observations, Wiley. 1984.

[18] C. Song, Z. Qu, N. Blumm, and A.-L. BarabÃąsi. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[19] L. Song, U. Deshpande, U. C. Kozat, D. Kotz, and R. Jain. Predictability of WLAN mobility and its effects on bandwidth provisioning. In *Proc. of IEEE INFOCOM*, 2006.

[20] H. Zeng, C. Ellis, A. R. Lebeck, and A. Vahdat. ECOSystem: Managing energy as a first class operating system resource. *Proc. 10th ACM ASPLOS*, 2002.

[21] Z. Zhuang, K.-H. Kim, and J. Singh. Improving energy efficiency of location-sensing on smartphones. *To appear in the Proc. of ACM MobiSys*, 2010.