Power Management for Stationary Machines in a Pervasive Computing Environment

Colin Harris and Vinny Cahill
Distributed Systems Group, Department of Computer Science, Trinity College Dublin.
email: Colin.Harris@cs.tcd.ie, Vinny.Cahill@cs.tcd.ie

Abstract

This paper examines effective power management of users' stationary desktop PCs in a pervasive computing office environment. The objective being to minimise the building's electricity consumption while maintaining acceptable desktop PC performance.

Current state of the art power management is focused on developing polices for mobile devices, which are ineffective for stationary machines. Effective stationary policies need to obtain context from the user of the machine, but there is a balance between what detail of context is necessary and how much this extra context costs both monetarily and energy wise.

We have implemented two location aware policies which detect presence of the user's Bluetooth enabled mobile phone. Trial results indicate that with these policies it is possible to get within 8% of optimal energy consumption with little performance degrade. However, this is the best case and the results are dependent on the user's usage patterns and the geographical layout of the office.

1. Introduction

With more and more computing devices being deployed in buildings there has been a steady rise in buildings' electricity consumption [9, 7]. These devices not only consume electricity, they produce heat, which increases loading on ventilation systems, further increasing electricity consumption. At the same time there is a pressing need to reduce overall building energy consumption. The European Union's strategy for security of energy supply highlights energy saving in buildings as a key target area [4]. Pervasive computing will potentially further increase the number of computing and sensing devices in buildings. How will this affect electricity consumption? In particular, what we are interested in is whether user context (derived from pervasive computing) can enable highly effective power management of stationary machines to significantly reduce the building's overall electricity consumption.

The end goal of our research is to develop a power management framework, which ensures all stationary machines in a building are effectively power managed. Effective power management ensures that the total electricity consumption of the building is minimised while maintaining user-acceptable service levels from the building's machines. In imagining an ideal case, all machines are instantly switched to low power standby states when not in use and these machines are restored to their operating states just before users request their service. For users to manually implement this policy is too onerous and some level of automation is required. To develop effective automated policies we need to obtain context from the user of the machine, in particular when the user is 'not using' the machine and when the user is 'about to use' the machine. Determining this user context is the most challenging part of the framework and there is a balance between how much energy additional context can save and how much it will cost both monetarily and energy wise. The ideal would be to leverage context that will already be available in the pervasive office environment. For example, estimated user location from wireless connections. The framework assumes the building's stationary machines can be power managed by software. The machines we are initially considering for the framework are desktop PCs, stationary laptops, video displays, photocopiers, printers, lighting and ventilation units. To explore the issues of this context aware power management, we examine in detail the power management of a user's stationary desktop PC. The objective is to minimise overall electricity consumption of the system while maintaining acceptable desktop PC performance.

The background section briefly reviews the current state of the art in dynamic power management and highlights the limitations of applying policies primarily developed for mobile devices to stationary machines. Section 3 details the requirements for effective power management of stationary machines and Section 4 discusses user-level policies for management of stationary machines. Section 5 describes two location aware power management policies, which derive location from the user's Bluetooth enabled mobile phone to power manage their PC. Results from the trial of these policies indicate it is possible to get within 8% of optimal energy consumption with little user perceived performance degrade. However, the results are highly dependent on user's usage patterns and the geographical layout of the office. Finally, the conclusions discuss potential improvements and the direction for future work.

2. Background

A general dynamic power management model, which can be applied to both mobile and stationary devices, is presented first to define common terminology. Then a brief overview of current research is given, concentrating on papers that report comparative data on power management policies for mobile devices. Finally, we discuss the limitations of applying power management policies developed for mobile devices to stationary machines.

2.1. Dynamic power management model

Dynamic power management [1] is a powerful technique for reducing device power consumption by taking advantage of idle periods during the operation of the device. The two fundamental assumptions are (i) idle periods will occur during the device's operation and (ii) these periods can be predicted with a degree of certainty. Figure 1 shows a graph of user requests and idle periods for a device over time (the dashed line). The power management policy (thick grey line) is deciding whether to power down in the current idle period. Some power management policies also attempt to power up the device just before the next user request.

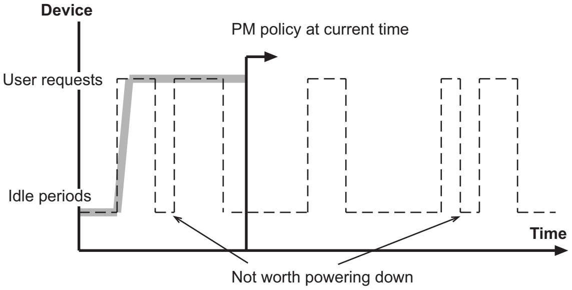

Figure 1. User requests and idle periods for a device

What makes dynamic power management difficult is the fact that for most devices power state transitions have a significant cost. Typically a power state transition may consume extra energy (e.g., spinning up a hard disk), reduce device performance (e.g., user waiting for device to wake-up) and possibly reduce it's lifetime (e.g., mechanical wear in hard disk spin-up). Therefore not all idle periods are long enough to justify powering down the device. The primary task of the power management policy is to predict (based on past information) whether the current idle period will be long enough to justify the transition cost. Secondarily, if the policy can predict when the next user request will be it can reduce the time the user is waiting for the device to wake-up.

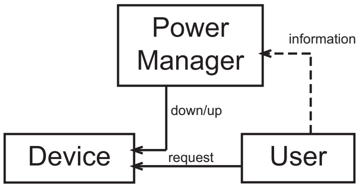

Figure 2. Dynamically power managed device

Figure 2 shows a simple model of a dynamically power managed device. The user generates requests that must be serviced by the device while the power manager implements policies that decide when the device should be powered down/up. Power management policies use information they receive from the user of the device to make their decisions. This information can be either observed or explicitly passed to the power manager by the user. The model can be viewed at different levels. For instance, the device could be a sub-component of a machine (e.g., a hard disk) or the machine itself (e.g., a desktop PC). Also, the user of the device can be viewed at different levels. For example, a low-level device driver, the operating system, a software application or an actual human user.

Each device can be modelled by a number of power states (S_0, S_1, S_2, S_3, ...). In the highest power state, S_0, the device operates at full performance. Subsequent lower power states operate at reduced performance levels. Either the device performance has been throttled and it operates more slowly or it is in a standby state. Each lower power state has an associated break-even and wake-up time (see Table 1). The break-even time (T_{be}) is the minimum time the device must be in the lower power state to amortise the cost of the state transition and the wake-up time (T_{wu}) is the time taken to transition back to the S_0 operating state. The deeper the power state the lower the power consumed but the greater the wake-up and break-even times. Typically standby state break-even and wake-up times are long relative to the corresponding times in lower power operating states.

The "oracle" policy [14] is a theoretical optimal policy which has future knowledge of user requests for the device. This policy will power down the device immediately after

Table 1. Power states, break-even and wake-up times

State	Power	Break-even	Wake-up
S_0	P_0	-	-
S_1	P_1	T_{be1}	T_{wu1}
S_2	P_2	T_{be2}	T_{wu2}
S_3	P_3	T_{be3}	T_{wu3}
...	...	...	...

a request is serviced to the lowest power state if the time to the next request is greater than T_{be}. The policy then powers up the device to the operating state just before the next request (see Figure 3). Since this policy powers up the device before the next request the break-even time only needs to consider transition energy and device lifetime, not performance degradation. This optimal policy is a useful baseline when comparing realisable policies.

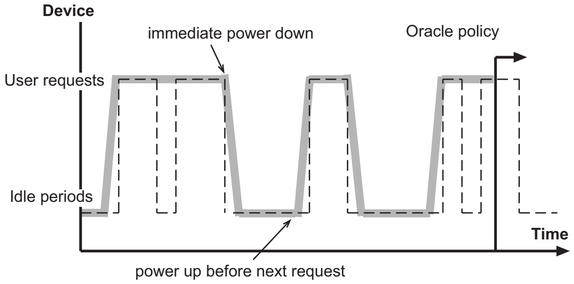

Figure 3. The theoretically optimal oracle policy

The key trade-off in the design of most real life policies is device power consumption versus device performance. Figure 4 shows a threshold policy that waits a given time T_{idle} before powering down in the idle period. It wastes energy waiting for the timeout and incurs a performance delay at the next user request. The shorter T_{idle} the more energy saved but the device will power down more often increasing the number of device response delays. There is also the added complexity that for some devices the transitions consume significant extra energy and/or reduce the device lifetime. Therefore an aggressive policy (with very short T_{idle}) could end up consuming more energy and/or cause the device to fail prematurely. We must also take into account the potential cost of implementing the policy. For example, extra energy may be consumed in the processor execution of a policy [14] or external sensor hardware may be needed, which will consume extra energy.

In the evaluation of policies suitable for stationary machines we must take all the above factors into account. In particular, we believe that users of stationary machines will not tolerate significant performance degrade as energy is not critical for these machines. Also, we believe a realistic solution will require a relatively low cost of implementation. Finding a power management solution that is transparent to the user with little additional overhead is key.

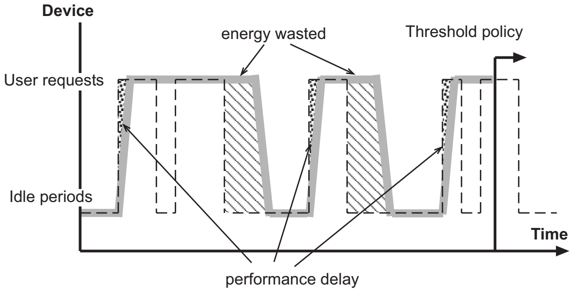

Figure 4. Trade off power consumption v's performance

2.2. Power management policy review

We conducted a detailed review of the current state of the art in dynamic power management. The objective was to gain insight into the area of power management and observe which techniques can be applied to power management of stationary machines. The review highlighted the fact that most research is applied to extending battery life for mobile devices with some research beginning in the area of power management for servers and server farms [3, 11]. During the review we noted that more advanced policies were using information from higher up in the system to make more intelligent power management decisions. We identified four power management levels (device-level, operating system-level, application-level, user-level) and categorised each policy according to the level it receives its information from.

The majority of policies are device-level and they concentrate on management of sub-components within the computer, either the hard disk, processor, memory or network card. These sub-components have relatively short break-even and wake-up times from their low power states with the hard disk having the longest.

Douglis [5] gives figures for two laptop hard disks. T_{be} (due to transition energy only) for the first is 5 seconds and 15 for the second and their wake-up times are 1.1 to 2.5 seconds respectively. He compares threshold policies (with different T_{idle} values) to the optimal oracle policy using a four hour usage trace of the hard disk for a machine which was running Microsoft Word and Eudora mail. The manufacturers recommended T_{idle} before spinning the disk down is 5 minutes. The oracle policy could reduce the 5 minute threshold policy's power consumption by 48% and the best threshold policy (with T_{idle} of 1 second) reduced the power consumption by 45%. This is within 6% of the optimum oracle policy. However, the performance degradation due to spin-up delays is very high. There was a total of 98 spin-up delays in the four hour trace period, one every couple of minutes. Douglis also notes that the performance of the threshold policy varies significantly depending on the usage trace and the performance characteristics of the hard disk. For example laptop hard disks have much lower spin-up times than those for desktop machines. Hence the same policy on a desktop machine would incur even worse performance penalties.

Lu et al. [13] have done a quantitative comparison of 11 different device-level policies from the simple threshold policy to predictive policies and more advanced stochastic policies. These polices are implemented for the hard disk of a desktop PC and are compared against the oracle policy and the worst case scenario of the disk being always on. Two eleven hour usage traces are used in the experiment, one developing C programs and the other making presentation slides. The algorithms are compared on power consumption, number of shut downs, number of wrong shut downs, average time sleeping and average time before shut down. The comparison shows that the time-indexed semi-Markov (SM), discrete-time Markov with sliding window (SW) and competitive algorithm (CA) are the best in terms of power consumption saving nearly 50% of power compared to the always on case and coming within 18% of the oracle policy. The CA policy is a special case of the simple threshold policy with T_{idle} equal to the break-even time [8]. The three policies are similar in power consumption but SW has less than half the number of wrong shut downs at 28 compared with 76 for SM and 64 for CA. The number of shut downs that occurred in the eleven hour trace for SW was 191 which equates to one every three minutes. Again this number of shut downs severely degrades the user perceived performance of the hard disk. The performance degrade of these policies is unlikely to be acceptable to the user of a stationary machine as energy is not a critical resource.

Operating system-level policies also concentrate on management of sub-components within the computer. We have only come across one research paper that has comparative data.

Lu's [12] task based power management (TBPM) policy uses the state of processes running in the operating system to find idle periods more accurately. For each device in the system the policy keeps a list of all processes using the device and its associated device utilisation. How soon the policy shuts the device down is a function of the total utilisation of the device. When a process terminates it is deleted from the list and when there are no more processes using the device it is shut down immediately. The policy includes a performance rule which ensures that no more than two consecutive shut downs are issued within time period T_w. Lu compares the TBPM policy with four device-level policies, exponential average (EA), event driven semi-Markov model (SM), competitive algorithm (CA) and threshold with T_{idle} of one and two minutes. The experiment was conducted on a personal computer with real usage traces for the PC hard disk. The results show that the average power used for TBPM was 0.435 Watts (W), SM was 0.507W and CA was 0.499W. These are the policies that perform best in terms of power consumption but the TBPM policy has far fewer shut downs due to the performance rule (181 compared to 477 for CA and 581 for SM). If we assume that the SM and CA policies are within 18% of the oracle policy (see above) then we can deduce that in this case the TBPM policy is within 2%. Lu claims that this additional OS level information enables the policy to find idle periods more accurately and hence can implement the performance rule without reducing the power efficiency. However, the device performance degradation is still large, 181 shutdowns in the 10 hour usage period, which on average is one shutdown every 3 minutes. Again, it is unlikely that users of stationary machines will accept this performance degrade.

None of the application-level policies we reviewed had comparative data to enable us to compare them with the device and operating system-level policies.

2.3. Discussion

All of the above policies focus on managing sub-components of the computer that have relatively short break-even and wake-up times. For example, a laptop hard disk's break-even and wake-up times are in the order of 10 seconds and 2 seconds respectively. Therefore the policies for these sub-components only need to predict short idle periods (order 10 seconds) and do not incur significant wake-up performance penalties (order 2 seconds). Another important factor is when these sub-components are off, they do not affect the visible state of the machine rendering it unusable for the standby period. The policies operate during the user's operation of the machine and balance machine responsiveness against extended battery life. The more aggressive the policy the slower the machine appears but the longer its battery life.

In general, the device-level policies reviewed above are aggressive with short timeouts and incur a significant number of wake-up delays (one every two minutes). Lu's TBPM policy uses higher-level information from the operating system but still incurs significant penalties (one wake-up every three minutes). These performance penalties are unlikely to be accepted by the user of a stationary machine as energy is not critical. There is no perceived gain to the user for suffering a less responsive machine.

To further compound the issue of reduced device performance, the gains from aggressively managing subcomponents of a stationary desktop machine are significantly smaller than powering off the entire machine.

3. Requirements for stationary machines

To gather requirements for power management of stationary machines we initially looked at the power management characteristics of the stationary desktop PC. This involved measuring the power consumption for each PC power state and the corresponding wake-up and break-even times.

Table 2 lists the measured power consumption for a DELL Optiplex GX260 running Windows XP (values are for the tower PC excluding the monitor). A Conrad Power Monitor Pro device was used to measure the PCs active power for each power state. On$_{max}$ is the power consumption for the machine at maximum load (i.e., CPU 100% and reading from the hard disk). On$_{average}$ was measured over a one hour period under typical load (user editing a document) and On$_{idle}$ is the power consumed with the machine idle (i.e., user logged on with no applications running). The table highlights the large drop in power consumption when the entire machine is put into standby compared to just switching the hard disk to standby.

Table 2. Power consumption of DELL Optiplex GX260

Device State	Power Consumed
On$_{max}$	87.3 W
On$_{average}$	48.4 W
On$_{idle}$	41.5 W
Hard disk standby	35.0 W
Standby	1.76 W
Hibernate	1.3 W
Soft off	0.8 W

Switching the hard disk to standby gives a 15.6% saving on the idle power consumption (41.5W), whereas switching the whole machine to standby gives a 95.9% saving. Modern PCs can achieve these very low power standby states by use of a dual mode power supply, which switches itself to a trickle mode when the PC is in standby [10]. Making this state change has significant benefits over just putting the hard disk into standby but also significant performance penalties.

The wake-up time from PC standby is 7.5 seconds and also, importantly, the transition to standby causes a visible change in the state of the machine rendering it unusable for the period. Making this state change when the user is still using the machine is clearly unacceptable.

To estimate the standby state break-even time we considered transition energy, device lifetime and performance degrade. To calculate the transition energy due to standby we measured the power consumption of a Dell GX260 tower PC over a one hour period. During this period the PC was manually transitioned on and to standby ten times. Software was used to record the exact time of transition and accurate measurements for On$_{idle}$ and standby power were obtained by measuring over one hour periods. Subtracting the On$_{idle}$ and standby power consumptions from the measured power consumption gives an estimated transition energy of 0.202 Watt hours (Wh) per transition. To make up this transition energy the PC needs to spend 17.7 seconds in the standby state. It takes 7 seconds for the PC to power down to 1.76W, therefore the break-even time due to transition energy is estimated at 24.7 seconds.

To estimate the break-even time due to device lifetime we assume the hard disk is the component most likely to fail first in the PC. The Sea Gate 80Gb Barracuda ATA V hard disk data sheet gives figures for the disks reliable lifetime to be $600,000$ power-on hours and $50,000$ start-stop cycles. The inverse of these figures give us a failure probability of $1.66 * 10^{-6}$ for every power-on hour of operation and $2 * 10^{-5}$ for every start-stop cycle. Clearly how the disk is used and power managed will determine its lifetime. We have taken the usage trace from the Standby On Bluetooth experiment below (see Section 5.1) to analyse its affect on the hard disk's lifetime. The total time spanned by the usage traces is 527.43 hours, with a total of 88.91 on hours and 87 standbys. Multiplying the on hours and number of standbys by their respective failure probabilities gives a total failure probability of $1.8 * 10^{-3}$ for the 527.43 hours. Extrapolating this probability to hard disk failure gives an estimated lifetime for the disk of 31 years. The current expected lifetime of a PC due to technology churn is from 5 to 10 years therefore the break-even time due to lifetime decay is negligible in this case.

We believe the most significant break-even issue for stationary machines is that of device performance degradation. If the policy does not power up the device before the user requests its service then the user perceives a delayed response time for the device. The user of a stationary machine is not prepared to suffer much waiting time for a device to be woken up as power is not a critical issue. Clearly this "performance" break-even time is more subjective as it is based on the user's tolerance of performance degradation versus energy saving. We estimate a performance break-even time for putting a desktop PC into standby to be in the order of minutes. The performance penalty is 7.5 seconds for the PC to come out of standby and the energy saving is 1 Wh per minute in standby (this equates to 0.01 cent per minute saving in monetary terms). We estimate a typical user would perceive the PC should be in standby for at least 5 minutes to make the performance penalty worthwhile.

So if the policy does not predict the next user request, it must be able to predict idle periods in the order of 5-10 minutes. If we can predict the next user request 7.5 seconds before it happens, our break-even time reduces to the transition energy and life-time factors which we have estimated to be 24.7 seconds. However, because our estimation methods are not detailed and to err on the side of caution, we give these factors a break-even time of 1 minute.

In conclusion, to achieve the significant standby power savings and not incur large performance penalties we must develop policies that can (a) accurately and quickly predict long idle periods (order of minutes) and (b) predict when the user will make the next request (order of seconds beforehand). The first requirement brings us close to the oracle policy in terms of power efficiency and the second requirement enables us to minimise performance degradation thereby approaching a power management solution that is both energy efficient and transparent to the user.

4. User-level context aware policies

To fulfil the above requirements of quick prediction of long idle periods and predicting the user's next request the two pieces of user context we need to know are (i) when the user is 'not using' the PC for an order of minutes and (ii) when the user is 'about to use' the PC an order of seconds beforehand (see Figure 5).

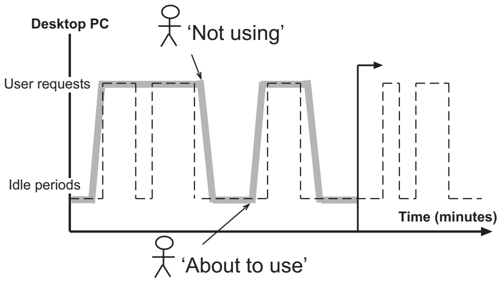

Figure 5. Required user context

Current user-level policies implemented are simple threshold policies that observe user requests from the keyboard and mouse input devices. However, this particular user-level information is not effective for accurately predicting the required user context. The presence of input events tells us that the user is 'using' the machine, possibly editing a document or browsing the web but an absence of input events does not tell us that the user is 'not using' the machine. The user could be reading from the screen or presenting a slide show to an audience. For this reason threshold policies are set to very long idle periods such as 20 to 30 minutes to avoid false shutdown of the device. Also, the threshold policy cannot predict when the user is 'about to use' the machine.

We conducted a simple case study measuring usage and consumption of a postgraduate student's desktop PC over a number of days. The student manually switched the machine to standby when leaving his desk and switched it back on when returning. The average daily power consumption was 410Wh compared to a calculated 530Wh consumed if using a threshold policy with idle timeout of 30 minutes. This demonstrates a potential 23% saving if we can implement a policy that knows when the user leaves and returns to his desk.

This policy can be automated by sensing the user's location. User location is a key piece of context for effective power management for two reasons. First, by detecting the user is not in the vicinity, we know for certain that he is 'not using' the PC and can power it down immediately. Secondly, by detecting the user returning, it allows time for the PC to resume before the user arrives and requests its use. Therefore we can detect when the user is 'about to use' the PC 7.5 seconds beforehand.

However, user location does not fully solve the problem as it is possible the user may not be using the PC when in it's vicinity. For example, the user could be sitting at his desk but in a meeting with a colleague, reading a journal or on the telephone. This user context, 'in meeting', 'reading', 'on telephone', could be useful for predicting the user is 'not using' their PC. Detecting this context is possible, but from close vicinity to the PC it will be very difficult to detect when the user will next use the PC 7.5 seconds beforehand. It only takes an instant for a user to turn around, look up and require use of the PC. Putting the PC into standby when the user is nearby will increase the number of short standby periods making it imperative to automatically resume the PC before the user needs it. Saving energy in these situations will be very hard to do transparently.

5. Location aware power management

To investigate the use of user location as a key piece of context for power management we have implemented location aware policies which sense the user's Bluetooth enabled mobile phone to determine their location. Mobile phones are an existing infrastructure and hence minimise the financial and energy costs of implementation. Also, we predict that mobile phones with wireless connections such as Bluetooth will become pervasive in the near future.

We have implemented two straightforward location aware policies and conducted user trials to evaluate their energy consumption and user perceived performance. The policies are the following:

1. Standby On Bluetooth (SOB). When the PC is on it polls for the user's phone via the Bluetooth discovery mechanism. If the phone is not found the PC powers down to standby. The user manually wakes-up the PC when s/he next requests it.

2. Standby/Wake-up On Bluetooth (SWOB). When the PC powers down to standby it passes control to a server in the room. When the server detects the phone again it sends a wake-up message to the user's PC (The reason we have used a sever to poll for wake-up events is because we have not found a Bluetooth device that can wake-up the PC from standby. We are expecting this functionality to be available in the near future).

We implemented the policies using the Winsock 2 Bluetooth API to communicate via a USB Bluetooth adapter with the user's Bluetooth enabled mobile phone. Using the Windows power management API we recorded all power state change events for the PC. This included when the PC was powered down to standby, when it resumed to the on state and when it was on but had been idle for the last minute. This on idle time enables us to estimate how much energy the policy wasted by the machine being on but (potentially) not being used. This gives us a measure of how much better we could do if we had more user context.

The range of the Bluetooth connection is 10 metres and its latency is approximately 10 seconds (i.e., it can take up to 10 seconds for the Bluetooth inquiry to find the phone) [2]. We also noted during implementation that sometimes the inquiry would not find the phone even though it was there. To overcome this source of error it was necessary to duplicate the number of inquiries to be sure the phone was not there before powering down. This polling process takes approximately 90 seconds to complete so there is a significant delay before the machine is powered down.

5.1. SOB policy results

We have conducted the SOB policy trial on four users, each for a one week period. The total usage trace time is 527.43 hours and the PCs were on for 88.91 hours and in standby for 438.52 hours with a total of 87 standbys. The policy is evaluated in terms of energy consumption compared to the optimal oracle policy and user perceived performance.

We estimated the energy costs of implementing the SOB policy by considering the energy consumed by the phone's Bluetooth radio and the energy consumed by the SOB policy polling the phone. For the phone energy we assume a base case of the radio switched off (this is the default setting). On average, the phone required one extra charge for the week period of the trial compared to normal use with the radio off. This extra charge consumes 6.25Wh of the building's energy and has been included in the policy comparison calculations. We measured the PC's power consumption when running the power management policy but there was no noticeable increase in energy consumption, therefore we have treated this as negligible.

We first examine the energy consumption for the best and worst case days and evaluate user perceived performance by anaylsing the standby period frequency of the entire trace. We then look at the energy consumption and performance per user trace.

In general the more often the user leaves the office during the day the more the SOB policy can save compared to the threshold policies. Figure 6 shows the power state graph for the best case user day which has many standby periods and no long idle periods (Note: In the graph, the standby periods are the horizontal lines at 1.76W, the on idle periods are at 41.5W and the on periods are at 48.4W). The SOB policy performs within 6% of the oracle compared to the threshold policies which range from 13% for the 5 minute threshold to 48% for the 30 minute threshold (see Figure 7).

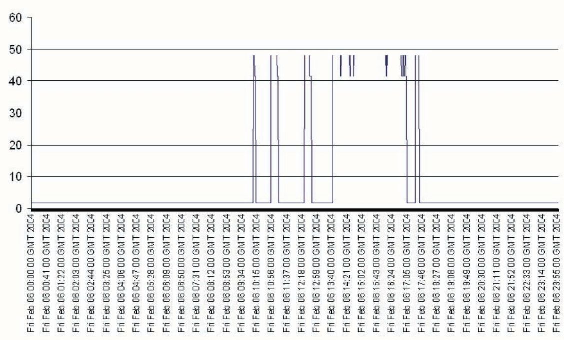

Figure 6. Many standby periods

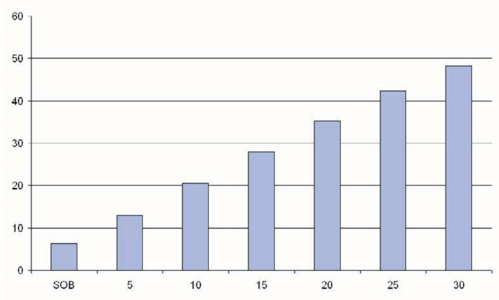

Figure 7. Percentage from oracle

However, long periods of the user not using the machine while still in the office wastes energy. Figure 8 shows the power state graph for the worst case user day with several long idle periods. The SOB policy performs within 27% of the oracle comparable in this case to the 25 to 30 min threshold policies (see Figure 9).

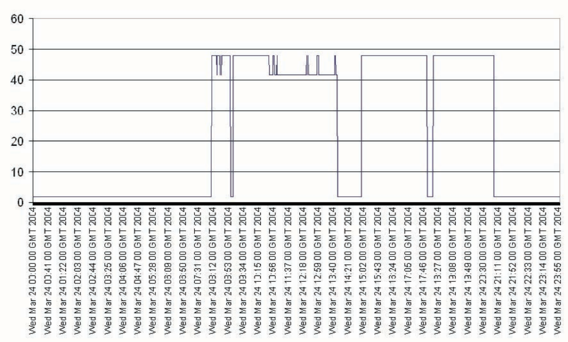

Figure 8. Several long idle periods

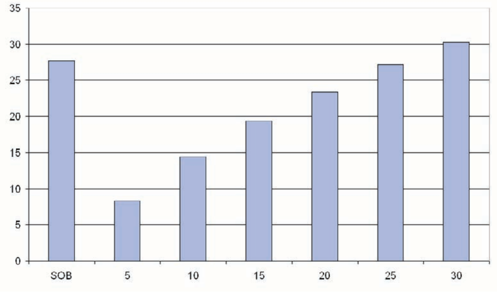

Figure 9. Percentage from oracle

The results show a significant number of short standby periods occurring, indicating users often pop in and out of their office for short periods. Figure 10 is a frequency graph of standby periods in one minute bins up to 30 minutes and all periods greater than 30 minutes. The graph highlights a significant number of standbys in the 0 to 10 minutes range (41%), which is the estimated performance break-even time for the PC. Therefore the SOB policy incurs a significant number of performance penalties that the user may deem unjustified. For devices which have significant lifetime decay (e.g., fluorescent lighting with estimated break-even of 5 to 10 minutes [6]) the policy would be in danger of reducing the device lifetime. The remaining majority of standby periods are greater than 30 minutes (47%).

Next, we compared the result data per user for their one week trace durations. Figure 11 shows the SOB and threshold policies' percentage from the oracle per user. It indicates that overall the SOB policy works well for some users but not for others. The best case users (A and B) come within 8% of the oracle while the worst (C and D) are 18% and 27% from oracle. Cases C and D are different, in case C the 5 min threshold performs much better than the SOB whereas in case D they are similarly bad. This is due to trace D having both a high number of standby and long idle periods.

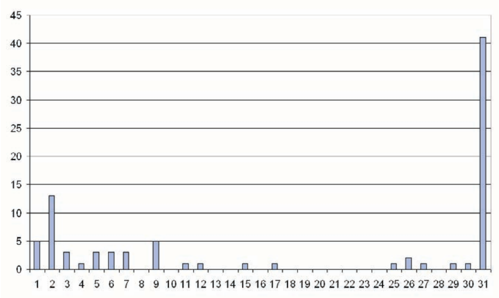

Figure 10. Standby periods frequency

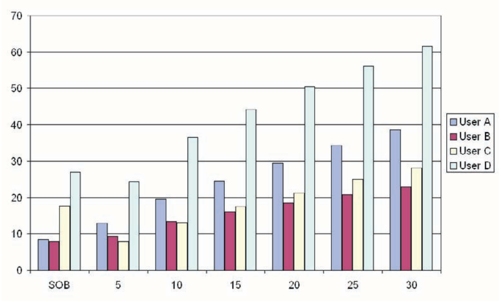

Figure 11. All users performance

To give more concrete numbers to the percentage figures above we estimate the financial cost for each policy over a year period for an office building with 800 desktop PCs and an assumed electricity cost of 10 cent per kilowatt hour (kWh). Table 3 details the total cost of the oracle policies per user and the extra cost of the SOB, 5, 15 and 30 minute threshold policies. For example, in the case of User A, the oracle policy costs 5,028 euro per year, the SOB policy costs an extra 401 euro whereas the 30 minute policy costs an extra 1,937 euro. In comparison, a policy of leaving the PCs on continually would cost 26,772 euro.

Table 3. Estimated policy costs in euro per year

User	Oracle	SOB	5min	15min	30min
A	5,028	401	645	1,236	1,937
B	6,453	480	598	1,034	1,476
C	6,446	1,133	510	1,124	1,813
D	2,920	765	711	1,290	1,796

The energy consumption of the SOB policy is dependent on how well the Bluetooth 10 metre boundary fits the user's working space. Users A and B work in small offices where the boundary fits the geography of their working space well. In other words they are heavy users of the PC within this boundary. The worst cases C and D work in open plan offices (see Figure 12) where other activities occur (e.g., meetings, experiment setups). In these cases the Bluetooth boundary does not fit their working space well and long idle periods occur when they are engaged in the other activities. A finer grained location sensing mechanism could adapt to the individual user's working space and thereby potentially avoid the occurrence of long idle periods.

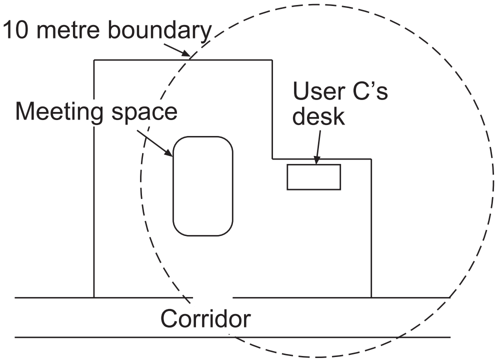

Figure 12. User C's office space

To evaluate the user perceived performance of the SOB policy for each user trace we use three indicators, the frequency of short standby periods (between 0 to 10 minutes), the maximum number of standbys that occurred in a one hour period and the maximum number of standbys that occurred in an eight hour period (see Table 4). User B suffers the highest number of short standbys (15) compared to the others (8 or 9). Users A, B and C have up to 7 or 8 standbys in a day compared to C which only has a maximum of 4 standbys per day. However, no user suffered more than two standbys in any one hour period.

Table 4. User perceived performance

User	Short standbys	1 hour	8 hour
A	9	2	8
B	15	2	8
C	8	2	4
D	9	2	7

We conducted a simple questionnaire of the users asking them what they thought of the policies performance. Users B and D said the response delay was not acceptable and thought it important that the PC wake-up automatically, while A and C said the response delay did not bother them and it would not stop them from using the policy. These results indicate the SOB policy's user perceived performance is borderline acceptable with some users requiring the PC to resume automatically.

5.2. SWOB policy results

We have conducted a separate one week trial of the SWOB policy for user A. In estimating the energy consumed by the policy we assume that the wake-up part of the policy is implemented on the user's PC (i.e., it does not require the server as was the case for the trial). Therefore, we have estimated the consumption to be the same as for the SOB policy. However, it may be slightly more if significant energy is required to power the Bluetooth radio when the PC is in standby.

Overall, the results are very similar to the SOB policy with no remarkable difference in the standby period frequency and coming within 8.7% from the oracle energy wise. The greatest number of standbys was also similar with no more than 2 in any one hour period and 7 in any eight hour period. However, the user perceived performance improved as the response delay was reduced from 7.5 to 2 seconds on average. This 2 second delay is a result of the long latency of the Bluetooth discovery mechanism (up to 10 seconds). Combining this delay with the wake-up time of the PC (7.5 seconds) implies that from entering the 10 metre Bluetooth boundary, the user's return to the PC must take greater than 17.5 seconds to avoid any response delay. Hence, the viability of the wake-up on Bluetooth mechanism is determined by the geographical layout of the office and the user's return path to their PC. User A's return path to the PC involves walking down a corridor (within the 10 metre boundary), through the office door and back up to the desk, which on average takes 15 seconds.

Clearly, a more real-time location sensing mechanism would improve the responsiveness of the wake-up policy and potentially eliminate all response delay from the SWOB policy.

6. Conclusions

Current power management policies developed for mobile devices are ineffective for saving energy in stationary machines. They manage sub components of the machine and trade-off significant performance penalties for relatively small energy savings. In general the biggest savings can be made by putting the entire machine into standby, but to do this policies must obtain context from the user of the machine. Current user-level threshold policies derive context from the keyboard and mouse input devices. This context is limited and long, inefficient timeout periods are required to avoid false power down of the machine. Location is a key piece of context for effective user-level policies as it tells when a user is 'not using' the machine and also when the user is 'about to use' the machine before it is requested.

The location aware SOB policy works well energy wise for two of the trial cases (within 8% of optimal). This is because in these cases the 10 metre Bluetooth boundary fits well to the users' working space and they are heavy users of the PC while in this boundary. The worst cases (18% , 27% of oracle) were for users in an open plan office where other activities occur within the Bluetooth boundary causing long idle periods. One possible solution is to use the Bluetooth radio signal strength to estimate a more fine grained location thereby potentially getting a better fit to the user's working space. However, switching the PC to standby when the user is still in the 10 metre vicinity will increase the number of short standby periods.

There was a significant number of short standby periods but there was no more than two standbys in any given hour of the user traces. The results of the questionnaire demonstrate the user perceived performance is quite subjective and that some users would require the PC to resume automatically. Another "performance" issue is due to the Bluetooth radio being always on, the user has to recharge their phone more often, which could be a potential barrier to adoption. Another barrier to widespread adoption is that not all users carry their phone with them when leaving the office.

The SWOB policy performs similarly energy wise and in specific cases is able to reduce the performance delay of the SOB policy. However, the long latency of Bluetooth discovery requires the Bluetooth radio to be situated along the return path of the user so it has enough time to wake-up the PC before the user arrives. This will not be possible for all office layouts. Clearly, a more real-time location sensing mechanism could overcome this problem. Another potential problem is the occurrence of false wake-ups, for instance in the case of user C, the SWOB policy might wake-up the PC as the user walks down the corridor past the office. Using more fine grained location could potentially overcome this issue.

We have carried out preliminary analysis of the BATs location system [15], which uses ultrasound receivers (located in the ceiling every 1.2m) to locate user Id tags. The accuracy of location is approximately 3cm with an estimated latency of 5 seconds. The estimated power consumption of the system for a 3,500 m^2 building is 13.8 kW or 3.9 W/m^2, which is considerable considering a typical office building consumes in the order of 40 W/m^2. However, a commercial development of this system using ultra wide band receivers has much lower consumption, only requiring four receivers per 200 square metre area. The accuracy is 15cm with a frequency of 40Hz (i.e., real-time). The estimated power consumption of this system for a 3,500 m^2 building is 552W or 0.15 W/m^2, which is 0.4% of the buildings energy consumption.

Future work will be the evaluation of fine grained location aware policies in terms of their cost, energy savings and user perceived performance.

References

[1] L. Benini, A. Bogliolo, and G. D. Micheli. A survey of design techniques for system-level dynamic power management. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(3):299–316, June 2000.

[2] J. Bray and C. F. Sturman. *Bluetooth Connect Without Cables*. Prentice Hall, 2001.

[3] J. S. Chase, D. C. Anderson, P. N. Thakar, A. Vahdat, and R. P. Doyle. Managing energy and server resources in hosting centres. In *Symposium on Operating Systems Principles*, pages 103–116, 2001.

[4] European Comission. Towards a European strategy for the security of energy supply, 2000.

[5] F. Douglis, P. Krishnan, and B. Marsh. Thwarting the power-hungry disk. In *USENIX Winter*, pages 292–306, 1994.

[6] T. Eino. Profitability of switching off fluorescent lamps: Take-a-break. In *RIGHT LIGHT 4*, volume 1, pages 113–116, 1997.

[7] R. Jain and J. Wullert, II. Challenges: environmental design for pervasive computing systems. In *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, pages 263–270. ACM Press, 2002.

[8] A. R. Karlin, M. S. Manasse, L. A. McGeoch, and S. S. Owicki. Competitive randomized algorithms for nonuniform problems. In *Algorithmica*, volume 11, pages 542–571, 1994.

[9] K. Kawamoto, J. G. Koomey, B. Nordman, R. Brown, M. A. Piette, and A. K. Meier. Electricity used by office equipment and network equipment in the U.S. In *Proceedings of the 2000 ACEEE Summer Study on Energy Efficiency in Buildings*. ACEEE, 2000.

[10] J. Kolinsky, R. Chary, A. Henroid, and B. Press. *Building the power-efficient PC*. Intel Press, 2001.

[11] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. W. Keller. Energy management for commercial servers. *IEEE Computer*, pages 39–48, December 2003.

[12] Y.-H. Lu, L. Benini, and G. D. Micheli. Operating-system directed power reduction. In *International Symposium on Low Power Electronics and Design*, pages 37–42. Stanford University, July 2000.

[13] Y.-H. Lu, E.-Y. Chung, T. Simunic, L. Benini, and G. D. Micheli. Quantitative comparison of power management algorithms. In *Design Automation and Test in Europe*, pages 20–26. Stanford University, March 2000.

[14] T. Simunic, L. Benini, P. W. Glynn, and G. D. Micheli. Dynamic power management for portable systems. In *Mobile Computing and Networking*, pages 11–19, 2000.

[15] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, Oct. 1997.