Increasing Predictability and Minimizing Power Consumption in KReSIT

Masters Project Stage I Report

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology

by

Bhavin Doshi
Roll No: 123059004

under the guidance of

Prof. Krithi Ramamritham

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
October 27, 2014

Contents

List of Tables

List of Figures

Abstract

For saving power at a building level, multiple techniques for reducing peak power is discussed and proposed in literature. The base power consumption forms major fraction of the consumption. To effectively reduce the energy bill, the reasons contributing to high base consumption should be investigated, and fixed. This report describes the data gathering, analysis and the actuation of fix done on a prototype lab in KReSIT. It also describes the need, feasibility and the proposed technique to scale up the solution at KReSIT level.

Chapter 1

Introduction

On IIT Bombay campus, the Department of Computer Science and Engineering is the second highest power consuming department, after the Department of Electrical Engineering. The electricity bill for our department is over Rs 1 crore annually. Given the scale of the expenditure, it was recommended that this expenditure be reduced.

Figure 1.1 shows the monthly electricity bill of the Department of Computer Science and Engineering at IIT Bombay. The data shown is obtained from the concerned administrative division in IITB.

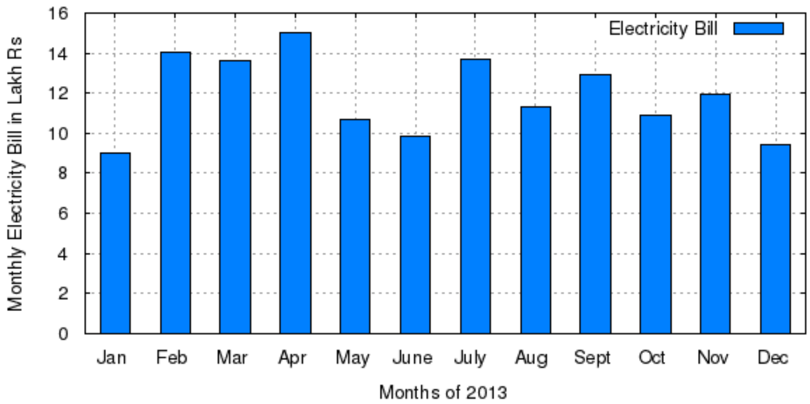


Figure 1.1: Monthly Electricity Bill of KReSIT

1.1 Power consumption pattern of KReSIT

To solve any problem, we need to first know its current state. To attempt saving power, we must first look at the current power consumption. There are two buildings in Computer Science and Engineering department - KReSIT building and Old-CSE building. There is already a smartmeter setup in the KReSIT building.

Figure 1.2 shows the power consumption pattern for a typical week for the KReSIT building as collected by the smartmeters. The total energy consumption is the area under the curve. Electricity bill is then directly proportional to the area under the curve. There are two components of the area - the base area, and the peak area. The area is dominated by the base power consumption, as seen in Figure 1.2. The base consumption is the minimum level of power consumption that is always present at any given time.

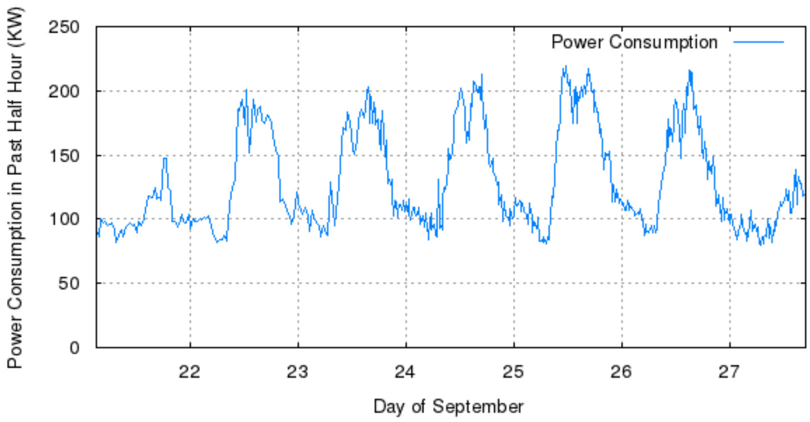


Figure 1.2: Power Consumption of KReSIT in September 2014

Many solutions focus on the peak power reduction, due to demand pricing scheme by the power distribution companies. The aim is to reduce the sharp spikes in power which lead to heavy penalty. Although these techniques are well positioned for the purpose of achieving consistent power consumption and reducing variability, they disregard the base component. The apparent repurcussion of a reduced base consumption is a reduced peak. However, sometimes a reduction in peak may not be observed indicating no wasteful appliance to be on, but this is hardly the case.

1.2 Components of base power consumption at KReSIT

To know if the base consumption is what it should be, we need to attribute the fractions of the base consumption to certain appliances that are using them. Power consumption at night in KReSIT around 4 AM is the lowest on any typical day, as seen in Figure 1.2, and can be considered representative of the baseload throughout the day. Total power consumption at 4 AM is 80 KWatts.

At the night, typically, there are corridor lights, desktop computers, servers, server room ACs and other miscellaneous devices running. 153 CFLs, 115 tubelights and 1 halogen light were estimated to consume 6 KWatts of power. There are also 7 ACs running in server rooms, with a total capacity of 18 tons, consuming 1.5KW per ton of power. These ACs would consume 27 KWatts of power. The ground floor server room's computing systems consumes 11KW. CC server room is expected to consume 23KW, based on the amount of equipment deployed there, servers and AC combined. Remaining 13 KWatts must be attributed to the miscellaneous devices and on-but-idle desktop computer systems. The proportion of the power usage can be seen in Figure 1.3. Since miscellaneous device consumption by routers, printers and phones is believed to be constant and necessary throughout the day, the focus for the base power reduction must go automatically to servers and desktops.

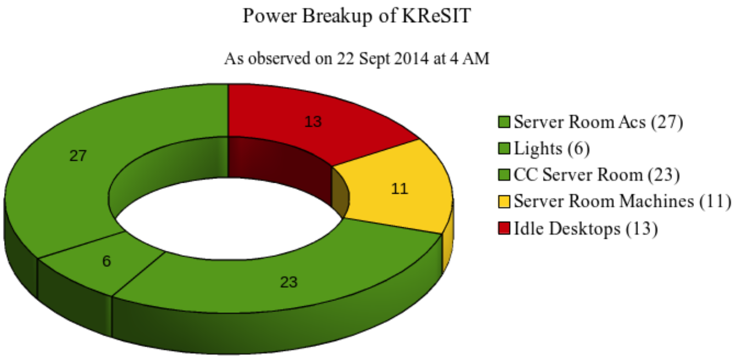


Figure 1.3: KReSIT Energy Consumption Breakup for 22 September 2014 at 4 AM

1.3 Summary

To summarize, the power wasted by servers and desktop computers is a clear target for reduction in the base power consumption. In subsequent chapters, an attempt has been made to assess the reasons behind the current situation and lack of corrective action taken on these points despite adequate information available on the problems themselves. The results from the implemented solution on a prototype lab in KReSIT and the proposal for a full scale imprementation of the same at the department level has been elaborated.

Rest of the report is structured as follows. Chapter 2 focuses on the power wastage by desktop computers and its remedies. Chapter 3 focuses on the power wasted by server computers and its remedies. Chapter 4 focuses on the additional information found while implementing these two solutions, and the implication of the new data. Chapter 5 presents the future work.

Chapter 2

Power Savings via Desktops

To reduce the base load, one of the focus points identified were desktop computers. The problem of power on machines in KReSIT has been known since quite some time. It was observed by many students and professors independently in the large labs that people leave their machines ON forever. People usually just switch off their monitors, lock the screen and leave their desk at the end of the work. At this level of operation, the system consumes around 45W of power, as measured by us in the lab.

2.1 Sensing Data

Since the problem is noticable at department level, the current work started from there, gathering more data at a finer level later.

2.1.1 Sensing department level machine count

Currently, there are 573 desktop computers in department. This estimate includes the department machines issued to MTech and PhD students, faculty and office staff, machines in NSL and OSL and the machines purchased from project fund by individual faculties. Table 2.1 shows the breakdown of the estaimate.

Category	**Count of available machines**
NSL and NSL-Annex	95
OSL	108
Private Machines MTech	130
Private Machines Faculty	44
Private Machines PhD	96
Misc other machines	Approx 100
Total	**573**

Table 2.1: Estimated Number of Non-server Machines in Department of CSE

2.1.2 Sensing the KReSIT level machine count

For the present work, the power consumption data of only KReSIT was available for further analysis, due to lack of installed smartmeter in old-CSE building. Hence, the NSL (95) and OSL (108) machines are not included in the count. Therefore, roughly 370 issued desktop machines excluding servers were considered.

An attempt was made to sense the total machines which were ON at around 4 AM, when the total power consumption is assumed to be representative of the base load throughout the day. At this time, the occupancy of the KReSIT could be assumed to be nearly zero. With the assumption that an ON machine will be connected and reachable on the network, a simple arp-scan in the subnet "10.129.0.0" (for 65536 possible hosts) revealed the ON hosts at 450. This included all network-connected devices, including printers, routers, phone and fax machines and smart powermeters and servers. All of these non-desktop devices were supposed to be always ON.

To get the count of these genuinely ON devices, a detailed analysis with a program called "nmap" to do host detection was done. This revealved that there are 92 devices which are non-computers. Out of remaining 278 computers, the IPs 10.129.1.* and 10.129.2.* are the department servers and the research project servers, and were counted to be 53. This provided the number of personal desktop devices to be around 295.

2.1.3 Sensing the NSL and OSL machine count

Even though power consumption numbers were not available for NSL and OSL, the machines were known to be running continuously, never shutting down or suspending. This was due to the department policy of keeping the machines available to users all the time. Hence, all 203 computers were running all the time.

2.2 Analysis

The main two areas identified while sensing were KReSIT and NSL/OSL labs. Each are analyzed separately below.

2.2.1 Analyzing the KReSIT data

In KReSIT, 295 out of 450 desktop computers were ON which had no human users to it. In other words, 65% desktops were left switched ON by their respective owners. With each desktop using around 45W, this resulted in power usage of 13.27 KW.

Students wanted their systems to be always reachable, so that the tasks like starting a long experiment remotely could be accomplished. They did not shutdown the machine, as switching it back ON remotely would not be possible. They also forgot to shutdown the machine, even if they knew that they would not need a remote access.

2.2.2 Analyzing the NSL and OSL data

NSL and OSL machines were always ON, which resulted in power wastage, as not all machines were always used by students, physically or remotely. At an hour where no one was using a machine, all 203 machines kept running wastefully, and consumed 9.13 KW power per hour.

In another observation, during a typical lab of the PG course CS699 (Software Lab), which is a compulsory first semester course for MTech students, it was observed that most students used their laptops in the lab. Due to this, in NSL, 68 desktops out of 95, i.e. approx 72%, were unused but were running, as observed manually during the lab. Clearly, they could have been in a suspended / shutdown state.

This observation is consistent with the data reported in literature[8][9]. However, these papers focus on savings power used by desktop computers in a corporate setting. Their solution also assumes a central control point for such machines.

Study of the same in the academic setting has not been done before. Such a study was only due, as the solutions plausible in corporate setting are also not suitable here. In the academic environment, a central control on machines is almost non-existant, as each student manages her own machine. This makes academic setting a new usecase for energy saving techniques.

Category	**Count of connected machines**	**Wasteful at 4 AM**	**Power wasted in an hour**	**Money wasted in 16 hours**[1]
NSL and OSL	203	203	9.135 KW	Rs 1461
Total machines in KReSIT	370	295	13.270 KW	Rs 2123
Servers	53	0	–	–
Routers, Printers etc	92	0	–	–
Total	718	498	22.405 KW	Rs 3584

Table 2.2: Estimated Power Wastage Due to Desktops In Department of Computer Science and Engineering

Table 2.2 summarizes these numbers for a quick overview. With daily monetary burden of Rs 3584, monthly expenditure due to the wasted power would be Rs 1.07 Lakh, with annual burden being at an estimation of 13.08 Lakh. To put things in perspective, with that money every year, 11 new RAs can be funded by the department, or 17 international conference trips can be sponsored, or 8 new 24core/16GB servers can be purchased.

2.3 Actuation of the solution

The following section first explains the technologies used to achieve an automated solution requiring least efforts from the user's side, by automating the parts of the solution.

2.3.1 Technologies used

There are two technologies used to implement the solution - suspend state of CPUs, and wakeonlan. Both are explained briefly.

Suspended state of a machine: All the modern desktop CPUs support a sleep state (S4) called suspend. In this state, the only component which gets power is RAM. Harddisk, CPU and network are taken off power. The RAM remembers the last state of the machine. The system can be resumed by hitting a key on keyboard. On resuming, the full system is powered ON, and the system resumes from where user left it last time.

Wake On LAN: The wakeonlan technology[1] allows users to remotely wake up a machine, without having to be physically around the machine. To do this, the user is required to send a special packet, known as magic packet, to a particular MAC address, and the NIC powers on the machine, in case it was suspended or shutdown. Simple commandline tools are available to send this packet with one command. This functionality requirs support of NIC of the target computer, which is usually the case for the machines not older than 5 years.

Wake Up on Packet Arrival: Some of the modern NICs support waking up the machine on any packet arrival, not just magic packet. In the present tests, it was found that this can result in no power savings, if the user's machine had a browser window with a GMail tab open. GMail javascripts keep exchanging heartbeats with server. Hence, when the machine goes down, it receives a heartbeat from the server within 10 seconds, and wakes up again. Therefore this option was not considered while building the solution.

2.3.2 Reducing power of machines assigned to students

Automatic triggering of power savings

Since many users forgot to suspend or shutdown their machine at the end of the day, power wastage happened. Expecting the user to perform some action to trigger the power savings might not always result in reduced power savings. The savings must be triggered automatically. Since the problem was power wastage, the solution was be to put the machines in a low power state, from where they would resume very quickly. However, this was required to be done automatically, without user intervention, for maximum gains.

All the major operating systems have the configurable policy to suspend the machine after a certain timeout of user inactivity. Inactivity is measured via absence of keyboard and mouse activity. Users must be forced by a policy to set this option, just once, to the value of not greater than 30 minutes. For the tests in the current study, it was found that this period is non-intrusive for all types of users.

It is then possible to sense if anyone has reverted it back, via the arp-scan of the network periodically. If the IP/MAC ownership is known, the person can be emailed about the non-compliance, and the savings can be re-enforeced. To handle the manual turn-off of the autosuspend policy, users can also be provided with an option to install an application to remind them via a popup note to again set the policy of autosuspend.

Ability to wake up the machine remotely

To honor the always-reachable requirement of the users, it was required that they be able to wake it up as and when required, with least efforts. The machine could be woken up with a keyboard-keypress when user was infront of the machine. To wake up the machine remotely, the technique of wakeonlan exists. However, magic packet sent in this technique is a MAC level broadcasted packet, and hence does not travel across the routers. This prevented users from using wakeonlan from outside the department.

To overcome this limitation, a server was deployed in the KReSIT subnet, with the functionality of wakeonlan being exposed via a webpage. Here is a brief description of the procedure. The users need to save their MAC addresses after a login. From anywhere in the world, they could then send this magic packet on the click of a button to the desired machine, and wake it up. Users were required to configure this on the wakeonlan server just once, after which they were able to use it from any device any time. It was found that the wake up time of the machine was also perceived to be non-intrusive by the users, as it did not involve any action on their part. The wakeup took around 8 seconds for the machines with a SATA/HDD drive, and around 3 seconds for machines with SSD.

Prevent the machine from sleeping when its being accessed remotely

The autosuspend policy works on the basis of no keyboard-mouse activity, hence a remote login session started by a user is disconnected after the autosuspend timeout. The OS is required to be made aware that a remote login connection is ON, and its not supposed to sleep.

This was achieved via a simple bash script. Most flavours of linux execute scripts in a certain directory before suspending. If any of the executed scripts return a non-zero status code, then suspend action is aborted, and the machine is kept ON. The implemented bash script simply checks for the existance SSH login sessions, and prevents the suspend if required.

2.3.3 Saving power wasted by machines in NSL and OSL

The solution begins with setting the suspend policy in each of the lab machine. The machines would suspend themselves after 30 minutes upon no user activity. The solution differs for the wakeup part.

Automatic Wake Up - Not Feasible

This could have been done transparently with a solution, where a programmable L2/L3 device would wake up the machine, if it detected a SSH session beginning. It turned out that to do this, the MAC address of the target machine must be hard-coded in the switch, which would be not maintainable. Also, the program would need to look at the L4 (transport layer) data of destination port to do this, which is also undesirable, as the device would step beyond its area of working.

First cut solution - Manual Wake Up

Given that it cannot be automated that easily, the first cut solution for NSL/OSL is to have a similar scheme as KReSIT. Since the machines are department owned,a public page would be created for each lab, containing the machine names and the wake-up buttons for each machine. Sysads would feed the MAC addresses of the machines to the wakeonlan-server as a one time setup, and would also update the same whenever the new machines are procured. Students using the lab would then be required to go to this public page, and wake up the concerned machine, before initiating a remote-login attempt.

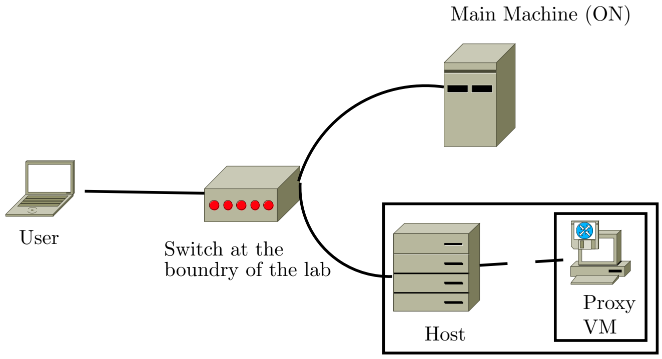
However, since deploying this solution required a policy change at the department level, we could not deploy it, and evaluate the gains.

Automatic Wake Up - With Virtualization

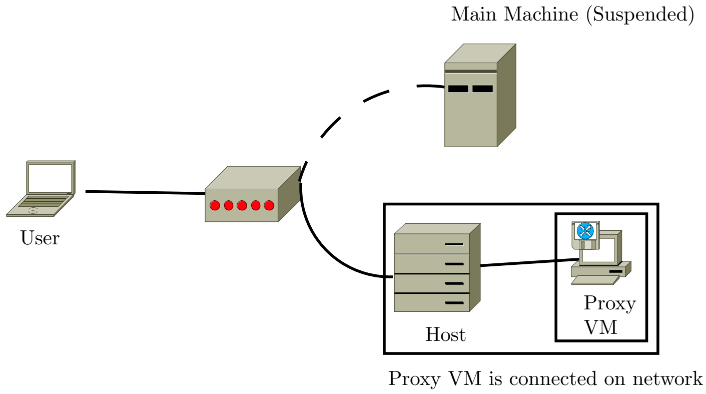
The transparent migration of desktops to save power is discussed widely in literature[3][4]. However, since the NSL and OSL desktops are physical desktops, and not VMs, applying the same techniques directly was not possible. Another set of techniques [10][9][8] discuss an approach where the sleeping physical machines are replaced by a proxy VM which responds to the requests coming in. However, these techniques require multiple components to setup, and are much more involved, with the benefit of covering more usecases. In case of NSL and OSL, the only usecase to be handled is the remote SSH login. In order to do the wakeup transparently, the following new lightweight method has been proposed[2], where tasks are accomplished with just a set of shell scripts on the user desktops, along with a virtualization-enabled host machine. The design of this solution is presented below, and the implementation is left to the next stage of the project, due to the required approvals from department.

[2] I came up with the technique independently, but while searching for similar work, it was found that a more complex version of it is already published[10]. Nonetheless, the implementation of the same remains to be done in the department.

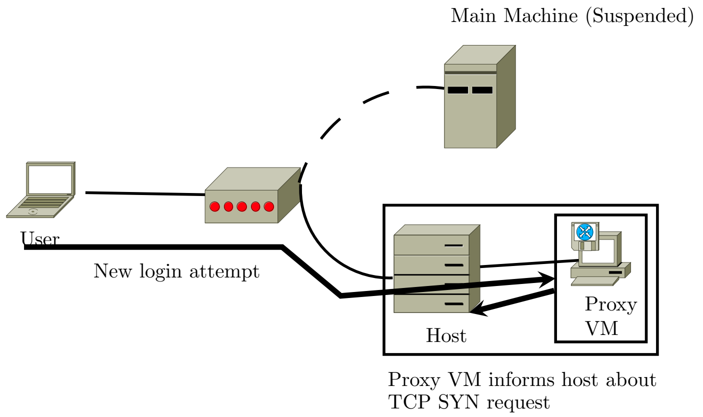
1. The solution involves creating a virtual machine with the same IP as the main machine. Call this the proxy VM. The host where the VM is hosted, is simply called host. Initially, the main machine is assumed to be ON. The proxy VM would be running, without a network connection.



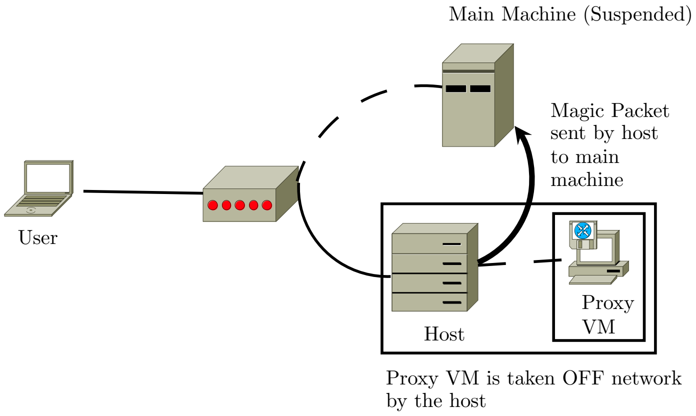
2. When the main machine decides to suspend itself, a bash script would intimate the host to connect the proxy VM on network. From then onwards, the proxy VM would receive any network request that comes for the main machine, as it has the same IP. Proxy VM would have a kernel module running, which would intercept all the incoming packets.



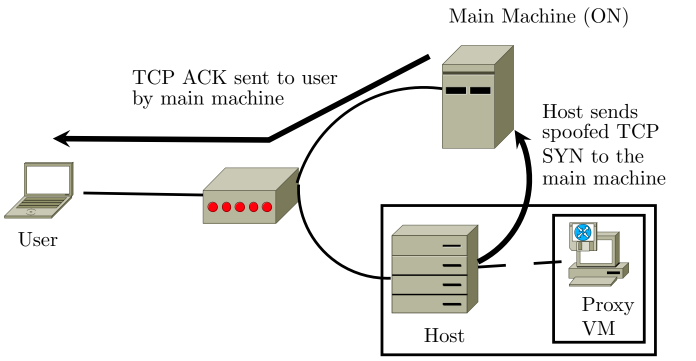
3. When user tries to initiate a remote login attempt over TCP, the TCP request will have destination port to be 22, in case of SSH. Similar ports for RDP or VNC can also be acted upon. Upon seeing such a packet, the proxy VM will inform the host of this special event.



4. When the host comes to know that the main system is required to be ON, so that the user can accomplish the remote login, it would first detach the network of the proxy VM. It would then send a magic packet to the main machine. Since all NSL/OSL machines have SSDs, they would take around 3 seconds to wake up.



5. When the main machine is up, the host will send a spoofed TCP packet, which will have source IP/port same as the packet received by proxy VM in step 3. Main host would then send a TCP ACK to the user directly, and the TCP handshake will happen between the main machine and user.



There are a total of 200+ machines in NSL and OSL, and for each one, a small VM instance of size 25MB RAM can be spawned from the same qcow image file. DSL (DamnSmallLinux) is a linux variant which can operate even with 16 MB of RAM. As it is desired that the VM responds to the SYN request coming at preconfigured set of ports, this is expected to suffice. This solution can hence be hosted on a normal desktop machine with 8 GB RAM and usual computing power.

Using this technique, a user will not be required to manually wake up the VM, and this is a huge usability improvement.

2.3.4 Prototype implementation and results

The solution for the KReSIT was implemented in a prototype lab, and results were promising. The SEIL lab was our prototype lab. It had two fulltime running desktop machines as servers, and 12 desktop machines. The implementation involved talking with users, and making them register their MAC address on the wakeonlan website. They were also asked to set a timeout for 15 minutes on their desktops. There was a smartmeter installed already, to measure the socket level load in the lab. Socket level power consumption of the lab before and after the implementation is shown in Figure 2.1. The implementation happend on August 12 afternoon, and drop in the base power consumption from 400W to 200W was observed from then onwards.

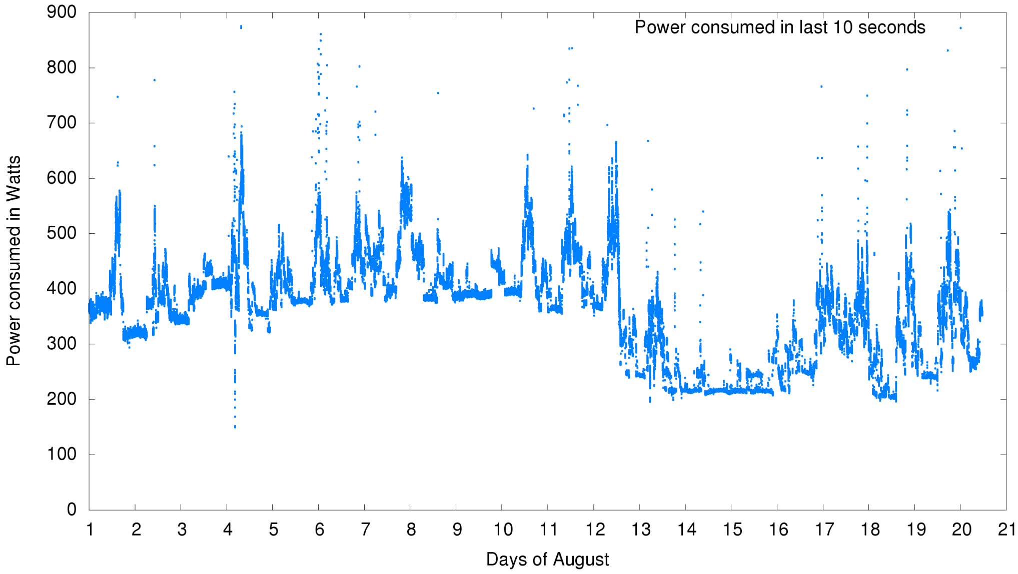


Figure 2.1: Socket Level Power usage of SEIL lab in month of August 2014

Out of 12 desktops in the prototype lab, 4 were always off. Out of the remaining 8, two desktops were old, and their linux deployment did not support waking up from suspend. Hence these two were also always ON. Power consumption by the these four ON machines was 200W, but the other 4 machines, which automatically suspended upon non-usage, ended up saving 160W of power hourly.

2.3.5 Expected savings from department wide implementation

If both NSL and OSL are covered under this scheme, then 203 machines will stop wasting power. Further, if KReSIT users are introduced to this solution, and are asked to use it, there can be varying degree of compliance. Depending on the fraction of people that actually implement this, the project annual benifit is depicted in the Figure 2.2. The calculation is done on the basis of Table 2.2, which reports that there are around 300 machines in KReSIT which are wastefully on for 16 hours a day in weekdays, and 24 hours a day in weekends.

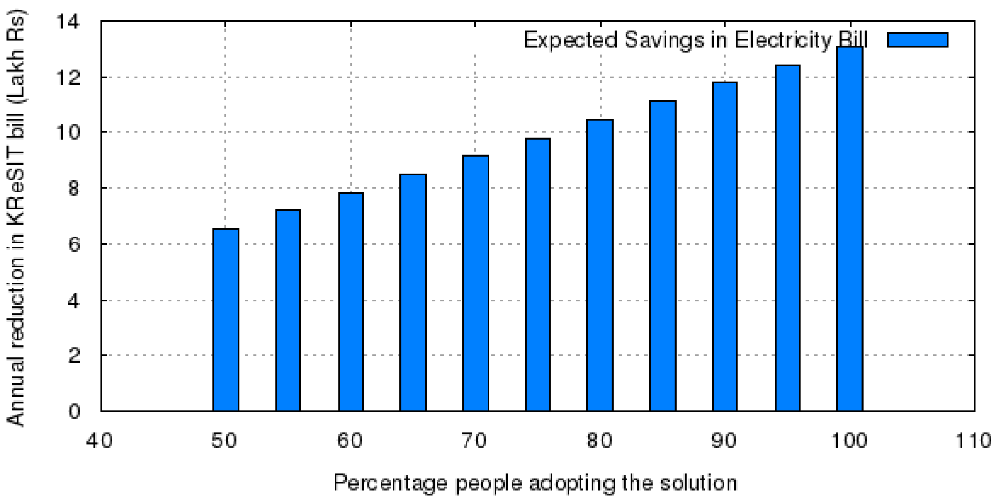


Figure 2.2: Expected monetary savings for varying degrees of adoption in KReSIT

Chapter 3

Power Savings via Servers

3.1 Need to consolidate servers

Power saving case can also be made for many small servers running 24x7, with very low load. Some examples are the personal websites of students and professors, group websites of various research labs and other data/backup servers. These have very low load on CPU, and are very lightly used, meaning that the utilization of CPU never goes above 20% for many[1] of the servers.

To save power, we need some strategy to bring the servers together and run on one hardware platform, so that they together consume lesser power. This task is in general called consolidation, and there are many approaches to it. Following subsection describes the approach, and its feasibility in the scenario at hand.

3.2 Ways to consolidate physical servers

The most primitive way of consolidating machines which are pure webservers, would be to copy all the files and configuration to one single machine, and let that serve all the web requests. It can be argued that since the servers are lightly loaded, they might have small-enough working set of files on the disk. So the approach here would be to create a VM with a vanilla OS, and then start re-creating each service that the server is offering. This can get very tricky sometimes, as the user is required to know every bit of the configuration for the server, and there is always a risk of missing out on a particular part of configuration which can break the functionality or compromise the security. The infeasibility of this approach comes from the potential risk exposure that the stakeholders get from the attempt of migration. Hence, even though this is cost effective, people rarely opt for this solution in real life.

Again, the fact that most servers are managed by a central team of sysads in a corporate environment makes it much simpler to migrate to an optimal deployment of servers. However, in universities, when each lab is managing their own servers, and when internal details of servers is not documented or known completely, it is next to impossible to do a whitebox migration of such instances.

If its not possible to go inside the contents of the machines, then they must be handled in a blackbox way. A way, which is agnostic to the contents and the configuration of the machine. Server consolidation[11] is a wellknown concept in the field of virtualization, with hypervisors being able to host multiple VMs, one for each server. It is also substantially motivated in the literature that server consolidation results in power savings[6][11]. However, the server consolidation is always studied in terms of existing VMs.

[1]It was difficult to obtain the actual count of such servers, as the department sysads too were clueless about the exact count of it.

This means that if we have p PMs and v VMs, we need to pack the VMs to PMs such that used machines are minimized, conditioned on the satisfaction of resource requirement of each VM[7]. The virtual disks are stored at a central location, and whenever required, the VMs can be migrated from one host to the other. This ofcourse assumes that all of the virtual disks can be stored at a central location.

However, consolidating a set of physical servers, with huge hard-disks can be a challanging task. In an academic environment, where the availability of resources like NAS is not always possible, the need to consolidate the physical machines still remains.

3.2.1 Approach 1: Virtualize All Resources, Black Box migration from PM to VM

The physical servers, when seen as blackbox, can be considered solely as a machine having CPU, disk, memory and networking resources. In order to virtualize the server, each of these resources can be virtualized. Hence, in case of consolidating 10 servers with 1 TB hard drive each, the space to store the full disk snapshot of these 10 servers, which is 10TB of space, would be required. Since this is a blackbox migration, there is no scope of taking limited snapshot of the disk.

Availability of such a high amount of storage on one box is usually a problem in the academic environment, and people end up keeping the individual servers powered on.

3.2.2 Approach 2: Virtualize All Resources, but Disk, Black Box migration from PM to VM

The second approach is to use the same physical disks in the virtualized environment. Xen hypervisor[2] allows the VMs to access the disk devices directly. We can plug-out the disk devices from their current server, and simply spawn a VM off them on a new machine with xen hypervisor. By doing this, the network, memory and CPU would have been virtualized, but not the disk. The VM would read the boot sector of the disk, and start exactly the way original machine would have started, without getting inside the internals of the server. Connecting multiple SATA drives to one motherboard can be achieved via PCI-to-SATA expanders available at low cost.

This way, since the disk is not on a central storage, it would not be possible to migrate the VMs off to a different host. However, this is not a problem since the servers are anyways lightly loaded, and the hotspot creation will be a non-probable phenomenon.

3.3 Prototype implementation and results

In the prototype implementation, there were two physical machines, hosting three websites. These two server machines were virtualized. Before virtualization, they both consumed around 45W of power individually, as measured in our lab. Post virtualization, the power consumption is reduced to half of total consumption. The whole physical to virtual migration exercise took less than an hour, and involved no reconfiguration of the servers.

Since all the IPs within KReSIT are portable, it was possible to take the servers from ground floor server room, and run them in the lab, with same IPs.

3.4 Expected savings from department wide implementation

All the labs in the department could be invited to be part of this migration, and free multiple smaller machines from running the lightly loaded servers. The task is achievable due to the blackbox nature of migration from PM to VM. The exact amount of savings in KReSIT is going to be not just the power saved, but also the machines gained from this exercise. Since there is always a need of getting experiment machines, the freed up machines will inadvertently be used for experiment purpose, maximizing the savings.

Chapter 4

Lab Occupancy Prediction From Network Connected Devices

To effectively reduce the power consumption of KReSIT, apart from fixing the issues causing the high base power consumption, the peak power consumption needs to be prevented. First step required in order to prevent the high base consumption and sharp peak power consumption entails knowing apriori what the power consumption might be.

4.1 Correlation between Power Consumption and Network-Connected Devices in KReSIT

In KReSIT, in order to sense the total number of connected devices on network, arp-scan was run every fifteen minutes. This provided the pattern of network-connected devices in KReSIT. This data had strong correlation with the power consumption graph of the KReSIT building. Figure 4.1 and Figure 4.2 provides a graphical representation of the same. The coefficient of correlation was 0.935 for the observed values, which indicates a very strong positive relationship between the two variables.

As power consumption is directly proportional to occupancy, and there is strong correlation between power consumption and network connected devices, one can be tempted to relabel the X-axis of the Figure 4.1 from "On machines" to "Occupancy". However, due to the wastefully ON desktop machines, the conversion from ON machines to occupancy is complicated. For a machine that is always ON, its difficult to say when the owner is or is not in the building.

Currently, the correlation graph gives us the slope of 700W. This means that when a user walks into the lab, she switches on many other things, most typically the ACs and the lights. When the first user walks into the lab, the increase in power consumption is going to be much more, which can be seen in the 500 to 600 users range in the Figure 4.1, where the points lie above the line, indicating higher consumption. But when more people walk in, there are no more ACs turned ON, and at the stable occupancy, the per person usage can be considered to be 700W.

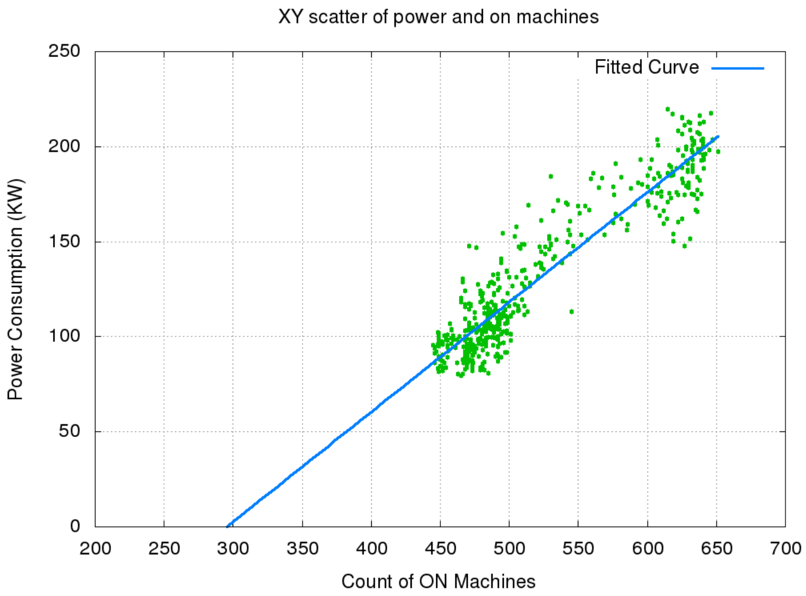


Figure 4.1: Correlation Between Power Consumption of KReSIT and Network-Connected Device in KReSIT

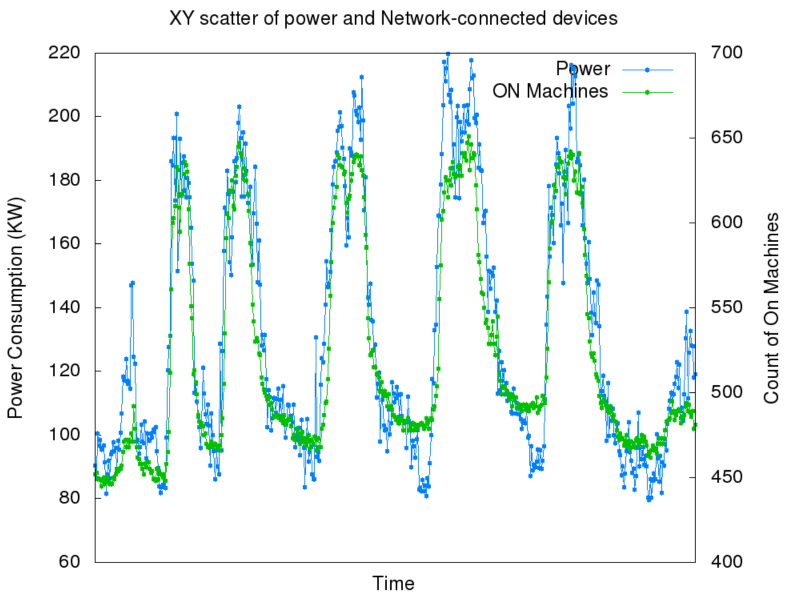


Figure 4.2: Plot of Power Consumption of KReSIT and Network-Connected Device in KReSIT

4.2 Improved Predictability of Power in KReSIT

Once the scheme for desktop power savings is implemented, the actual occupancy in the lab can be inferred via the number of network connected machines. This has positive implication on the predictability of the total power consumption in KReSIT. Figure 4.3 summarizes the implication.

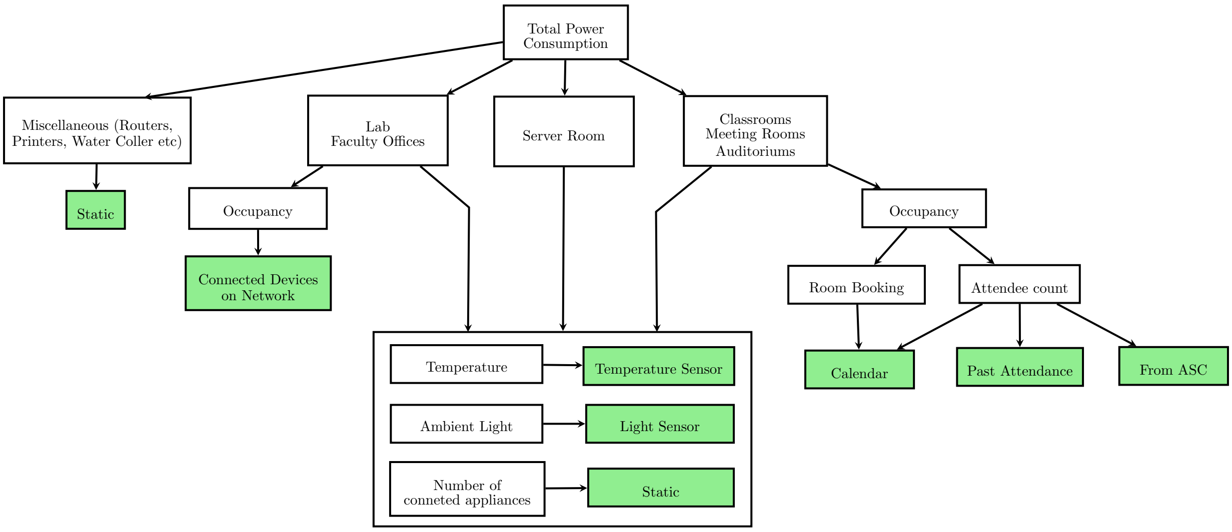


Figure 4.3: Dependence of Power Consumption on Various Factors In KReSIT

Total power consumption of KReSIT depends on four components - occupancy of labs and faculty offices, lecture/seminar/auditorium halls, server rooms and miscellaneous consumption. The miscellaneous consumption by printers, routers and scanners etc. shall remain constant, and is statically computable. The server room consumption depends on the computer deployed there, along with the number of ACs deployed. It also depends on the ambient temperature outside. The lecture hall power consumption depends on the calendar. Even if one lecture has just 5 students, and the other 50, the total consumption by the classroom is going to be predictable (either half or full classroom capacity). This consumption very lightly depends on occupancy, and definitely does not depend on the network connected machines, as far as KReSIT is concerend. Since the lab power consumption is the only one that depends on occupancy, a strong correlation between the powered on machines and the total consumption was obtained.

Hence, the implementation of the scheme suggested in Chapter 2 for KReSIT desktops do not solely have the power savings benefits, but enables a new way to measure occupancy, at the building level, without requiring any additional sensing equipment. Even though this is specific to the computer oriented academic and corporate buildings, this is a significant achievement, as the current occupancy detection schemes involve very complex mechanisms sometimes[5]. This sensed data can provide ground truth of the current occupancy, which can aid in predicting future occupancy. This prediction would eventually lead to predicting power via a parameterized model.

Chapter 5

Conclusions and Future Work

5.1 Novel Contributions

Apart from the proposed scheme to save power in the department, two other major contributions are as follows.

- Discovered that the total KReSIT power consumption is directly proportional to the number of networked connected devices, and hence one of the significant variables in the power model to be built.
- A lightweight way to achieve continuous network presense of a machine for limited set of connected oriented services, while saving power by suspending the machine, without actually virtualizing the machine at all.

5.2 Work to be done in Stage 2

Future work in the next stage of the project includes following points.

- Implementing the desktop power saving solution in KReSIT
- Implementing the first cut solution of lab system power savings in NSL and OSL
- Implementing the server system power savings via virtualization at KReSIT level
- Implementing the transparent modifications for power savings in NSL and OSL
- Expanding the desktop power savings idea to most of the departments in IIT, including the academic and administrative offices. Here the implementation would trivially involve setting the auto-suspend timeout, and nothing else, for most of the systems in offices, as users do not do the remote login, and the usecase of waking up the machines remotely is not applicable. However, estimating the power savings from this source is not possible right now, as the count of machines across IIT, and data about energy wastage owing to them is not available immediately.

Bibliography

[1] Wake on lan. http://en.wikipedia.org/wiki/Wake-on-LAN.

[2] The xen project, the powerful open source industry standard for virtualization. http://www.xenproject.org/.

[3] N. Bila, E. de Lara, K. Joshi, H. A. Lagar-Cavilla, M. Hiltunen, and M. Satyanarayanan. Jettison: Efficient idle desktop consolidation with partial vm migration. In *Proceedings of the 7th ACM European Conference on Computer Systems*, EuroSys '12, pages 211–224, New York, NY, USA, 2012. ACM.

[4] T. Das, P. Padala, V. N. Padmanabhan, R. Ramjee, and K. G. Shin. Litegreen: Saving energy in networked desktops using virtualization. In *USENIX annual technical conference*, 2010.

[5] A. De Paola, M. Ortolani, G. Lo Re, G. Anastasi, and S. K. Das. Intelligent management systems for energy efficiency in buildings: A survey. *ACM Comput. Surv.*, 47(1):13:1–13:38, June 2014.

[6] R. Koller, A. Verma, and A. Neogi. Wattapp: An application aware power meter for shared data centers. In *Proceedings of the 7th International Conference on Autonomic Computing*, ICAC '10, pages 31–40, New York, NY, USA, 2010. ACM.

[7] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo. Dynamic resource management using virtual machine migrations. *Communications Magazine, IEEE*, 50(9):34–40, 2012.

[8] S. Nedevschi, J. Chandrashekar, J. Liu, B. Nordman, S. Ratnasamy, and N. Taft. Skilled in the art of being idle: Reducing energy waste in networked systems. In *NSDI*, volume 9, pages 381–394, 2009.

[9] J. Reich, M. Goraczko, A. Kansal, and J. Padhye. Sleepless in seattle no longer. In *USENIX Annual Technical Conference*, 2010.

[10] Y. A. S. Savage and R. Gupta. Sleepserver: A software-only approach for reducing the energy consumption of pcs within enterprise environments. *Power (KW)*, 100(150):200.

[11] S. Sundarrajan and H. Nellitheertha. Server consolidation and virtualization. 2006.