Green Operating System:
Future Low Power Operating System

Anshul Pachouri
Student, DDM
Jaypee Institute of Information
Technology
Sector 62, Noida

Mohit Sharma
Student, B.S
Jaypee institute of Information
Technology
Sector 62, Noida

Tribhuwan Tewari
Senior Lecturer (Department of
Computer Science)
Jaypee Institute of Information
Technology
Sector 62, Noida

Prashant Kaushik
Lecturer (Department of Computer
Science)
Jaypee Institute of Information
Technology
Sector 62, Noida

ABSTRACT

Energy Consumption has become a critical issue for all kinds of computer systems, data centers and servers. Their demand for 24X7 connectivity and availability needs huge power in quantity of megawatts. The performance of microprocessors has been improving at an exponential rate and this trend is continuing from past two decades. However, increased performance does not come for free. One of the most important consequences of higher performance has been a dramatic increase in power consumption. For example, Intel 386 processor initially consumed about 2 Watts of energy; a Pentium 4 can use as much as 55 Watts. Various algorithms have been developed with respect to present needs and situations, but no one thinks of the future.

This paper discusses the novel approach to reduce the power consumption of the operating system and gives an entirely new dimension to look in. We have given the name "green" to that operating system, as its more energy efficient. Our paper will also explore the future power issues of the operating system and the holistic approach to tackle them, in the form of an entirely new operating system viz. Green OS.

Categories and Subject Descriptors:[Operating system]: Low power operating system

General Terms

Algorithms, Management, Documentation, Performance, Design Standardization, Reliability, Theory, Verification.

Keywords

Low Power, Green OS, Localized Power Shutdown, Cloud Computing

1. INTRODUCTION

The day by day increase in the number of systems has tremendously increased the power consumption from the past. We have made number of operating system related utilities, based on different functionalities, but now there is a greater need of a low power consuming operating system. A typical Linux server gulps about 225 Watts of power, meaning that the millions of working Linux servers (around 27% of market share) are responsible for nearly 5 million tons of carbon emissions annually. Furthermore, Springboard research recently reported that an average-size server has the same carbon footprint as a mid-size four-wheel- drive vehicle. So, there should be an approach which should focus on to reduce the power consumption in desktops. It might not save much power as far as a single PC is concerned, but on a broader scale, it saves a huge amount of power. Our work would also be extended to mobile Operating systems, so that battery charging becomes a weekly business, instead of a daily routine. There is a lot of research done previously by various researchers working on low power consuming hardware but lesser work has been done in the area of developing low power operating system. The launch of Big Green initiative by IBM, in August 2007, had a goal to help its clients integrate Linux into an enterprise to reduce costs and energy consumption by building cooler data Centers. Big Green Linux is a part of Project Big Green, which is another initiative to reduce energy consumption both for its data centers and clients. Apart from IBM, Intel is another organization which is working in area of green innovations in the computer science.

2. Literature Survey

It has been observed that there are times when the processor is in stall mode that is waiting for the data from the memory. Processor stall can be used to increase the throughput by temporarily switching to a different thread of execution, or

reduce the power and energy consumption by temporarily switching the processor to low-power mode [1] which can be represented as aggregating the processor free time for low power consumption.

Till date, processing speed of applications is not fast enough but the number of applications are increasing day by day; so, it is better to switch the processor between different operating modes keeping in mind the trade-off between low power and high performance. There are some processors like Intel XScale [2] and Transmeta Crusoe [3] which makes it possible for the frequency of the processor to be reduced, with proportional reduction in voltage.

Voltage scaling and slowing down of frequency has to go hand in hand else the power saving will be an offset by equal increase in execution time, which will lead to no reduction in the total energy. Since, energy is directly proportional to the square of the voltage; reduction of operating voltage can yield considerable energy savings [4].

The central issue with processors whose performance can be changed is how the right level of performance can be obtained. The goal is to reduce the performance of the processor without causing an application to miss its deadlines (see figure 1). Completing a task before its deadline and then idling is less energy efficient than running the task more slowly to begin with, and meeting its deadline exactly [5]
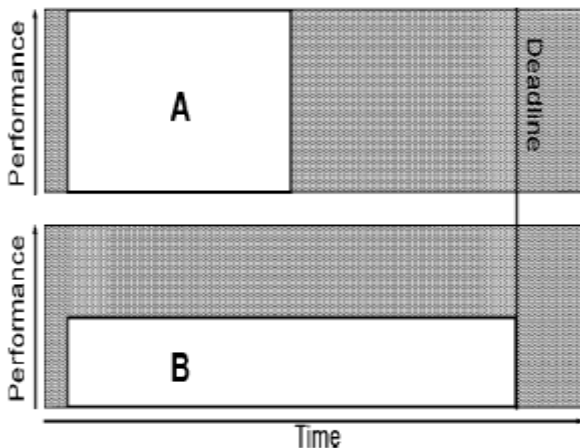

Figure 1: B: process running in low power state, A: process running normally

Hardware-supported Dynamic Power Management (DPM) provides a mechanism to save disk energy by transitioning an idle disk to a low-power mode. However, the achievable disk energy saving is mainly dependent on the pattern of I/O requests received at the disk. In particular, for a given number of requests, a bursty disk access pattern serves as a foundation for energy optimization [6].

The original warp processor design was primarily performance-driven and did not focus on power consumption, which is becoming an increasingly important design constraint. Focusing on power consumption, an alternative low-power warp processor design and methodology that can dynamically and transparently reduce power consumption of an executing application with no degradation in system performance is introduced. This helps in achieving an average reduction in power consumption of 74% [7].

Likewise, traditional exploitation of instruction level parallelism saturates as the conventional approach for designing wider issue machines leads to very expensive interconnections, big instruction memory footprint and high register file pressure. New architectural concepts targeted to the application domain of media processing are needed in order to push current state-of-the-art limitations. To this end, we regard media applications as a collection of tasks which consume and produce chunks of data. The exploitation of task level parallelism as well as more traditional forms of parallelism is a key issue for achieving the required amount of MOPS/Watt and MOPS/mm2 for media applications [8].

Multiprocessor Systems on Chips (MPSoCs) leads to undesirable power consumption characteristics for computing systems that have strict power budgets, such as PDAs, mobile phones, and notebook computers. SuperCISC is a heterogeneous, multicore processor architecture designed to exceed performance of traditional embedded processors while maintaining a reduced power budget compared to low-power embedded processors. At the heart of the SuperCISC processor is a multicore VLIW (Very Large Instruction Word) containing several homogeneous execution cores/functional units. In addition, complex and heterogeneous combinational hardware function cores are tightly integrated to the core VLIW engine providing an opportunity for improved performance and reduced energy consumption [9].

Most of the research done by various people in the field of low power Operating system is based on some assumptions. For instance, one has done voltage scaling by taking in account that we will work for particular set of application such as multimedia, some of them have taken account for type of application like interactive, non interactive, periodic or non periodic in terms of resources etc.

It is been argued that running all hardware components at all time is not necessary .We must think of way that we can shut down some of our components(Ethernet card, audio chipset ,etc) in such a way that energy is saved and user efficiency is also not affected (which are in use at time or which can scheduled)

3. Future Power Operating System Requirement

The number of computer and related systems will be greatly increased in the coming future as there would be an increase in demand due to the growing population with higher disposable income. Hence, the power consumption will also be rapidly increased.

The Internet has penetrated into our daily lives to a large extent. Today, much of the computation goes online like performing calculations, converting file into different formats, even reading, writing and saving files online is possible by the medium of Google docs. Recently the introduction of Chrome operating system by Google brought the operating system, itself online. Analyzing the trend, it can be easily seen that the whole computation would be online which would be supported by increasing bandwidth, enhanced internet architecture and supporting infrastructure. This will give rise to the abundant

usage of cloud computing, which is basically defined as paradigm of computing, in which dynamically scalable and often virtualized resources are provided as a service over the Internet. In simpler terms, in near future, neither setup would be required nor would anything be required to be installed in our computers; everything would be online; just an internet browser would be necessary to cater all your computations and system requirements.

There are higher possibilities that the hardware size would also be greatly reduced as to support just one application, but this discussion is out of scope of our paper. Hence, networking applications and hardware would be the first concern when we talk about the low power consumption. When the computation and every other utility will be online, then the operating system would definitely see the use of networking applications in the computers. This will also increase the number of network computing severs across the globe and hence, the power consumption issues would be based on two broad areas: networking applications and network servers.

The network based operating system would require the low power consumption techniques which are efficient for running networking applications. We have to think and focus on optimizing the efficiency of the networking hardware and reduce its power consumption.

The techniques such as aggregating processor free time [1], Voltage scaling [4] are being implemented for multimedia applications. There is very less work in area of making low operating system with reference to the networking applications. The algorithms which have been building so far are able to solve the existing problem of power crisis, but none of the algorithms have been developed considering the future. After a decade when the system would change, then again they would have to design other algorithms for optimizing kernel and other components for less power consumptions. That's not all; only few of them have worked on optimizing the power consumption of the network cards.

4. Proposed Methodology for Future Operating System

The future operating system will be network based and would be efficient in running networking applications via cloud computing. The different methods which can be used to make future low power operating system are access based file prediction, localized power shutdown, CRAMES (Compressed RAMs for embedded systems [10], Low power Warp processors, Tickless Kernel and virtualization.

Access based file prediction means to pre-fetch the files which will be required by the operating system and the user based on the previous requests and access. As, the operating system is network based, analyzing the different request made the user will give us his access pattern and hence we can pre-fetch those files which save our power and improve performance a lot.

Local power shutdown refers to component shutdown depending upon the prediction of requirement of various components of computer. The decision of keeping the system idle or shutdown is taken on the basis of calculating the tradeoff between system uptime and keeping it idle. This can be implemented by two ways.

4.1 Layout of Schema 1

1. Calculate the power of each device ex: Ethernet card, hard disk, CPU etc on various modes: ideal, busy, sleep.

2. Schedules hardware component via scheduling algorithm without affecting the efficiency of the user.

3. Proceed on the internal structure and will work on finding the energy use by per set of instruction. For instance, in case of CPU, run it for 5 minutes in an infinite loop .After that calculate the total energy and total no of instructions. Divide them to calculate energy per instruction.

4.2 Layout of Schema 2

1. Make a graph of the power consumed and number of instruction computed per unit time.

2. Try to locate all the zones of the graph where no of instruction have crossed the optimum range of computations.

3. Schedule the processes according so that both requirement are satisfied i.e. no change in efficiency and considerable change in power.

Another point to be noted here is that we would work mostly on cloud computing so we not need hard disk or any other storage media at every moment of time. So we will put it in sleep mode till time we are just doing the computation process and use it at the time of saving any data.

CRAMES and warp processors are very good options to reduce the power consumption at the hardware level. CRAMES, an efficient software-based RAM compression technique for embedded systems. The goal of CRAMES is to dramatically increase effective memory capacity without hardware design changes, while maintaining high performance and low energy consumption [10].

4.3 Tickless Kernel

In this kernel the programmer have modified timer tick, now we will have timer ticks when they are needed. The end-user benefit is cooler-running processors and increased power savings. On experimentation they we have found significant changes in desktops and notebook PC.

Tickless kernel/time resource is to reduce power consumption during CPU-idle periods, by causing the CPU to go 'totally' idle, rather than going 'idle-with-ticks'.
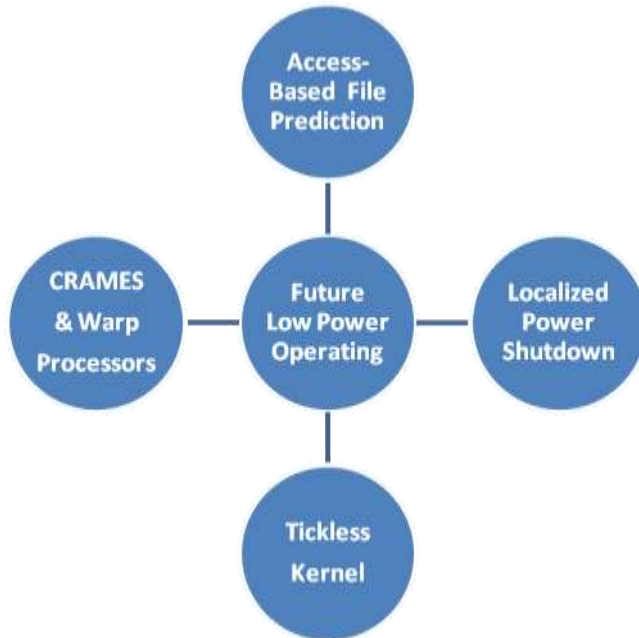
Figure 2: Model for Future low power operating system

5. Conclusion and Future Work

The problem of the low power operating system rose because in our past we never thought that the problem of power consumption would arise. In near future, most of the applications would go online by using cloud computing. So we would work and focus mainly on optimizing the power of the network applications and system at both the levels: end user and server level, so that history would not repeat itself.

The present methods are very much efficient in saving the power of the operating system but still lot of research is going on in this area. There is a great need to address the power issues which are going to be raised by the future systems considering the growth in the number of the systems being produced now. Energy would be the key issue tomorrow as per the present conditions of the power resources in developing and underdeveloped countries.

The future work of this paper is the implementation of the localized power shutdown and access based file prediction technique in the Linux Kernel. Further, it may also be implemented at the hardware l e v e l if sufficient funds are available.

6. REFERENCES

[1] Aviral Shrivastava, Eugene Earlie, Nikil Dutt, Alex Nicolau .Aggregating Processor Free Time for Energy Reduction. *CODES+ISSS'05,* Sept. 19–21, 2005, Jersey City, New Jersey, USA.

[2] Intel 80200 processor based on Intel Xscale micro architecture.

[3] D. Laird, Crusoe processor products and technology manual (January2000)

[4] T. Pering, T. Burd and R. Brodersen, Voltage scheduling in the ARM microprocessor system, Proceedings of the international Symposium on LowPower Electronics and Design 2000 (July 2000).

[5] Krisztián Flautner, Steve Reinhardt, Trevor Mudge Automatic Performance Setting for Dynamic Voltage Scaling 7th International Conference on Mobile Computing and Networking, 2001 Rome Itlay

[6] Caching for Bursts (C-Burst): Let Hard Disks Sleep Well and Work Energetically, ISLPED '08: Proceeding of the thirteenth international symposium on Low power electronics and design.

[7] Low-Power Warp Processor for Power Efficient High-Performance Embedded Systems DATE '07: Proceedings of the conference on Design, automation and Testing.

[8] J. P. Robelly, H. Seidel, K. C. Chen, G. Fettweis Energy efficiency vs. programmability trade-off: architectures and design principles. 2005 ACM Transaction Design Issues.

[9] Alex K. Jones, Raymond Hoare, Dara Kusic, Gayatri Mehta, Josh Fazekas, John Foster. Reducing Power while Increasing Performance with SuperCISC. August 2006 ACM Transactions on Embedded Computing Systems (TECS)

[10] CRAMES: Compressed RAM for Embedded Systems September 2005 Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/software codesign and system Synthesis .

[11] Low Power Operating System Project (Lesswatt.org by Intel)