Power-Aware Operating System for Interactive System

Vinaya. I. Math
Dept. of Computer Science and Engineering
Sai Vidya Institute of Technology
Bangalore, India

S. Priyanka
Dept. of Computer Science and Engineering
Sai Vidya Institute of Technology
Bangalore, India

Anagha. S. Dixith
Dept. of Computer Science and Engineering
Sai Vidya Institute of Technology
Bangalore, India

Surabhi Karanth. V
Dept. of Computer Science and Engineering
Sai Vidya Institute of Technology
Bangalore, India

Abstract— **Power management saves the power by shutting down the idle devices. These devices often serve requests from concurrently running tasks, ordering task execution can adjust the lengths of idle periods and exploit better opportunities for power management. As a increasing awareness of economic and environmental cost of power consumption by interactive systems, has raised an important issue to design not only hardware but also an operating system on an energy efficiency scheme. However, architectural techniques is not the only process which helps in the power reduction but it includes some more procedures. This paper presents an approach of operating system to reduce power consumption. This can be done if the operating system kernel estimates the utilization of a device from each process. If a device is not under use from any of the programs ,the device can be put into lower power state .This paper gives how the scheduling can facilitate power management .This paper presents an approach for using OS to reduce power consumption of input and output devices in laptops ,pc's (interactive systems).**

Keywords — Interactive systems; Operating system; Scheduling; Flexible time; Caching;

I. INTRODUCTION

Operating system is an intermediate software between user of a computer and computer hardware .Due to rapid advance in hardware system supports variety of application to support these applications, Operating System play an important role .The computer components are hardware and software . Hardware components form the resources of a computer. Software defines the way in which these resources are used.

Functions of operating system are:

1) USER INTERFACE:

Basic function of operating system is to provide convenient environment for the user.

2)PROGRAM EXECUTION:

Operating system needs to load and execute the program.

3)INPUT OPERATION:

User cannot directly control directly control input devices therefore operating system provides a means to interact with this.

4)RESOURCE ALLOCATION:

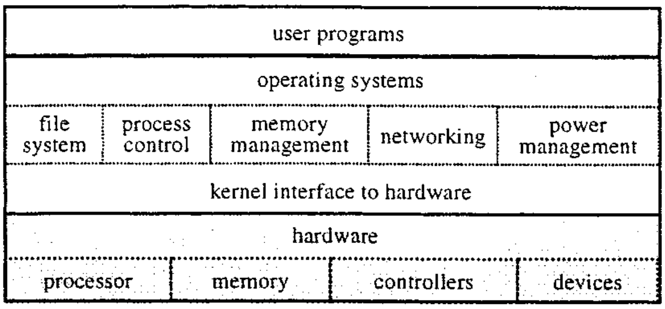
When multiple users or multiple jobs runs at the same time operating system manages the resources and allocates such as CPU, main memory and file storage.

Operating system acts as the coordinator between system software, hardware and application software.

Power is also a precious resource hence it needs to be managed. High power usage raised the environmental impact so power reduction is of higher importance [1].and a field needs investigation for optimality.

II. BACKGROUND

Fig 1: LAYERS OF COMPUTERS



user programs
operating systems
file system | process control | memory management | networking | power management
kernel interface to hardware
hardware
processor | memory | controllers | devices

It shows layer architecture of conceptual computer bottom layer is for hardware components such as memory , processors and input/output devices. Operating system communicates with this layer through BIOS (basic input output system calls) ,BIOS are privileged instructions .User programs access hardware resources through system calls.

Fig2 , Shows the life of a process. The process is created, it will run and finally it gets terminated, in between it makes the requests for the hardware services.

when a user starts a program a process is created .This process occupies memory and takes CPU time .It may also read files from the hard disk and write the files to the hard disk ,even operating system supports multiprogramming, where many process share resource and execute concurrently. Normally used scheduling algorithms for the process management are FIRST IN FIRST OUT,SHORTEST JOB FIRST,PRIORITY SCHEDULING MULTILEVEL QUEUE SCHEDULING[5].

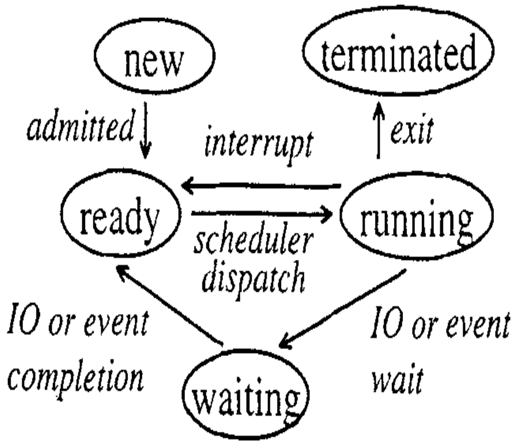


FIG 2:SHOWS PROCESS LIFE CYCLE[5].

So operating system acts as abstractor and manager .As a manager operating system need to save power and should define an effective means for power utilization.

Power reduction techniques are of two types:

1.Static power reduction technique.

2.Dynamic power reduction technique

Static techniques are applied at design time(compilation time),power gating is one of the static power reduction technique ,this is effective but they cannot guarantee run time based variations.

Dynamic techniques are applied at run time,

These dynamic techniques are called as dynamic power management(DPM)

DPM techniques include clock gating.

Clock gating is a technique or set of techniques used to reduce dynamic power consumption .It has been implemented in many commercial products such as strong ARM processor[3].

Strong ARM processor provide DPM in three modes:

1)Normal mode(full on mode)

2)Idle mode(power down mode)

3)Sleep mode

These are the DPM techniques are implemented in processors ,but its implementation for input and output devices are different from processors as input and output devices has low power sleeping state(idle) and high power working state[4].

1: DYNAMIC POWER MANAGEMENT (DPM):

No computers operate continuously .Some of the devices of the computers are busy where other devices are idle. When devices are in idle state it provides an opportunity for power reduction [4].

Dynamic power management creates a virtual effect that the devices serves readily for requests ,even though the devices sleep thus creating power down state and reducing power usage.

2: BREAK EVEN TIME:

It is the minimum time length of an idle period of a device to save power [6] it depends on configuration of devices but it is not dependent on scheduling request policies .so it is necessary to use break even time as a possibility factor for power reduction

3: CONCEPT OF JOBS AND POWER MANAGEMENT:

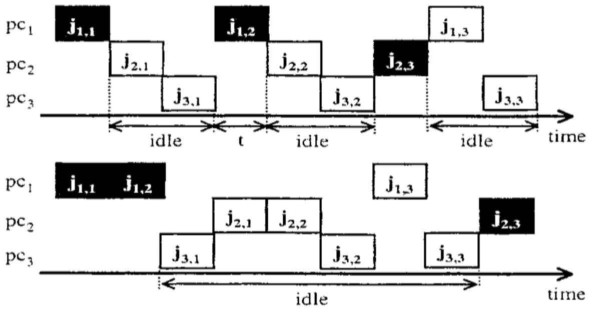


FIG 8:TWO SCHEDULES OF THREE INDEPENDENT PROCESSES:

Job is a unit to finish the specific task, it can be scheduled to complete its assigned task in specific time ,here pc1,pc2 and pc3 are processors, each process has three jobs ,so there are nine jobs to schedule, in these nine jobs j1,1 must be executed before other two jobs of pc1,this is called as precedence constraints. Another constraint is the deadline constraint .If the job finishes after or before the deadline it can cause idleness of a the input output devices. So , it is necessary to schedule the jobs for power management. Fig:3 shows 2 schedules, here black boxes indicates this job needs the device .I One difference between the two schedules is length of the idle periods. In schedule 2, the idle period is continuous and long, So, the operating system saves the power in schedule2 is long hence here power management algorithms can save power.

4:POWER SCHEDULING IN INTERACTIVE SYSTEM

In personal computers ,laptops a user can work on text editor mean while he can read a mail content and store this content on hard disk .If these two requests are not arranged in a proper manner the disk get short time idle periods. If they are arranged such that they arrive at same time the disk(hardware resource)can sleep for longer duration it means if jobs on interactive system are scheduled properly than there is chance of power reduction

RELATED WORK

Power reduction is based on operating system. Here operating system observes the co-ordination between the process and the devices and estimates the power utilization for each device.

1) These policies are of 3 types:

Timeout, predictive and stochastic the complete study is available in [7] and impact can be studied in [8][9].

Timeout policy: It is based on how long the device will remain in a particular state before the transition decision is taken. It is also based on the workload .

Predictive policy: It is the prediction of idle period of a device before it starts its job. If the device is not doing the work for longer time that its idle time will be long so that it can shutdown and save power[1].

Stochastic policy: This is the optimal way to adjust the speed and the mean response time in the processor to reduce the power consumption. It is used in models like Petrinets and non stationary model.

TO APPROACH REDUCE POWER IN MEMORY ALLOCATION

The input output devices consume power. Addition to that memory also consumes power .So, a handsome amount of power can be saved by using power performance modes at compilation time and run time.

Page allocation affects the power consumption i.e it is necessary to provide a sophisticated hardware component such that the pages are easily accessible so thereby we can reduce the power consumption [5].

Memory access should be in such a way that the most frequently used pages should be easily accessible, so that CPU should consume less time to access those pages and there by the power usage is reduced

BLU(Bulk Loading and Unloading) is one of the best methods. It is the ability to analyze the data without fitting the entire data set into the memory.

APPLICATION PROGRAMS FOR LOW POWER CONSUMPTION

Application programs should be modified to balance between the OS and available power ,but we have to keep in mind that the quality of the service should be up-graded and at the same time the performance should also be balanced, But it requires further study to develop a set of API's, few API's that can be used are:

imp.sleep(),imp.deepsleepfor(),server.sleepfor(),server.sleeputil(),imp.wakeup().

POWER MANAGEMENT FOR PROCESS

a)REQUEST GENERATION MODELS:

Requests are generated by process. While execution different processes consume different amount of power .So, process gives us a valid information about the idle periods of devices .A process will be in the running state if it is selected by a process scheduler[5].

Studies discover that process have short lifetime. Process which requires more input output utilization often have shortest lifetimes. Ex: ftp and gcc. But these programs oftenly requests.

On the other hand daemons like syslogd have long life time. But they rarely request for input output utilization. So it is necessary to know about the termination of the process.

If we will get to know the termination time of the process, we can schedule such that the device sleeps for more time, and we can design the power down state of a input-output device so that the power consumption of the device can be reduced.

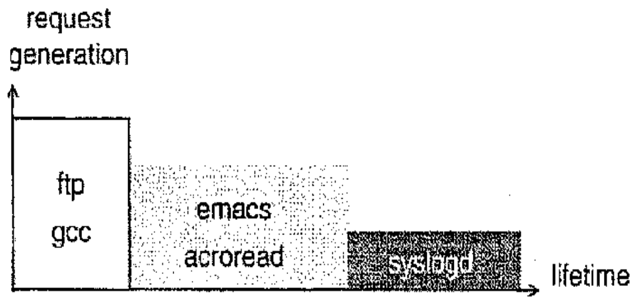


FIG4:PROGRAMS THAT PRODUCE REQUESTS HAVE SHORT LIFETIME.

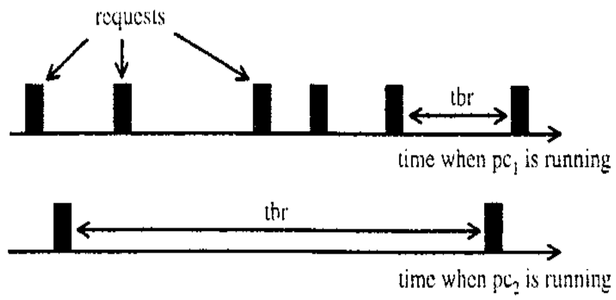


FIG5: TIME BETWEEN REQUESTS OF TWO PROCESSES.

So, these power reduction techniques can be implemented by the following three things that is:

1) It should know about the termination of the process.

2) It should consider the number of times the process executes.

3) Utilization of devices for power reduction.

POWER REDUCTION ON BASIS OF TYPES OF PROCESS

Processes are of 3 types:

1) CPU burst: Process making the CPU computation more.

2) I/O burst: Process using I/O devices more.

3) Process between CPU and I/O burst.

So, we can estimate the duration of time when the device is not doing any useful task [busy]. This idle time can be defined as the time between completion of previous requests and the arrival of new requests.

So, it is necessary to develop the algorithms to shut down the devices in this idle time and save power.

SHUT DOWN CONDITION

A device should be under the shut down condition if its total usage belongs to minimum length of the idle period. Shut down condition can be determined by "aggressiveness factor".

Approach should be stressed on individual process that its waiting state should be useful to make a long sleep for a input – output device.

Shut down faces two problems:"how to shut down" and "when to shut down". A device should shutdown only if its idle period is minimum, if its less then minimum the process of shutting down is a overhead procedure,

We can opt the Conventional shut down approaches or Rest mode

OFFLINE SCHEDULING

Offline scheduling is performed before the execution of a job , it is only possible if we have complete knowledge of job requirements in advance

Online scheduling is best for interactive systems. But we need to have the background of the offline scheduling.

Offline scheduling resolves complex constrains but does not supports flexibility it means it provides limited flexibility offline scheduled real time systems use total band width server in a guaranteed available bandwidth s but it does not provide effective solution for power management.

. process creates jobs at run time so, scheduling must be performed online to save power because through offline scheduling it is impossible to predict appropriate behavior of the process and the other point is offline scheduling of multiple devices is a complex problem.

ONLINE SCHEDULING

Scheduling process at run time eventually helps to determine the idle time of devices and helps in power reduction

a) Request created by timers:

Request arrival time from process is some what predictable. In Unix a job is performed by the call back function is scheduled by timer,

If power manager knows additional information about scheduling done by the timer then manager can save power more effectively.

b) predictive shutdown:

straight forward shutdown methods describes improvement in energy but it decreases computer responsiveness. So, we need to explore a mechanism to shutdown the system where we try to predict the idle time length based on computation history and shutdown the input output devices if the predicted length of idle time make no loss to responsiveness.

b) Predictive wake up:

If a process requests a sleeping resource that the process should wait for wake up delay to perform predictive wake up accurately the scheduler has to know which devices are used by the jobs. This can be done by two ways.

1) Predict using the history of a process but the disadvantage is low accuracy.

2) Providing an interface for a process to give their device requirement advantage is only requested device wakes up disadvantage is programs need to be modified.

Power reduction can also be implemented by flexible timers and caching.

IMPLEMENTATION

Power management is implemented in Red hat 6.2 Linux on the Sony VA10 note book.

LENOVO BEIJING POWER MANAGEMENT SYSTEM:

Lenovo's energy management interface tools allow a user to modify the settings to make better use of available energy. It is preinstalled software manufactured by Lenovo. It is necessary to be overviewed to make it more specific for input output devices of interactive systems.

CONCLUSION

This gives an approach to reduce power consumption in interactive systems. Algorithms should be included to exploit the opportunities of long idle periods.

Study can be extended in several directions to balance performance with the power awareness . Software techniques for power aware OS should be investigated further.

Finally it is a paper which discusses the topics of low power management of multiple devices.

ACKNOWLEDGEMENT

The authors are greatefull for the comments of reviewers, this publication is based on the thoughts collection of Yung-Hsiang Lu ,Luca Benini, respected members of IEEE.

REFERENCES

[1] M. B. Srivastava, A. P. Chandrakasan, and R.W. Brodersen, "Predictive system shutdown and other architecture techniques for energy efficient programmable computation," *IEEE Trans. VLSI Syst.*, vol. 4, pp. 42–55,Mar. 1996.

[2] J. M. Rabaey and M. Pedram, Eds., *Low Power Design Methodologies*. Norwell, MA: Kluwer, 1996.

[3] Crusoe Processor. [Online]. Available: http://www.transmeta. com/crusoe/

[4] R. Golding, P. Bosch, and J. Wilkes, "Idleness is not sloth," in *Proc. USENIX Winter Conf.*, New Orleans, LA, Jan., 1995, pp. 201–212.

[5] Operating system concepts by Abharam Silberschatz,Peter B.Galvin Greg Gagne.

[6] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Trans. VLSI Syst.*, vol. 8, June 2000.

[7] G. De Micheli and L. Benini, "System level power optimization: Techniques and tools," *ACM Trans. Design Automation Electron. Syst.*, vol. 5, no. 2, pp. 115–192, Apr. 2000.

[8] Y.-H. Lu, E.-Y. Chung, T. ˇSimunic´, L. Benini, and G. De Micheli, "Quantitative comparison of power management algorithms," in *Design Automation Test Europe*, Paris, France, Mar. 2000, pp. 20–26.

[9] Y.-H. Lu and G. De Micheli, "Comparing system-level power management policies," *IEEE Design Test Comput.*, vol. 18, pp. 10–19, Mar.–Apr. 2001.