IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified

Vol. 6, Issue 3, March 2017

Energy Optimization Factors for the Embedded Mobile Processors

Vrushabh Khobragade[1], Mandar Gadkari[2], Shubham Shingne[3], Hamza Niaz[4], Shivam[5], Dr. Prakash Prasad[6]

Research Scholar, Computer Science & Engineering, Priyadarshini Inst Engg. & Tech., Nagpur, India [1,2,3,4,5]

Professor, Computer Science & Engineering, Priyadarshini Inst Engg. & Tech., Nagpur, India [6]

Abstract: Due to the increasing performance of smart phones the users are expecting more exorbitant performance from the company. As the performance increases the usage of battery consumption also increases and the battery drains very rapidly. To overcome the problem we use the software technology, to increase the battery life and to efficiently run any tasks without any fear of the battery drainage of the mobile rapidly.

Keywords: Embedded Mobile Processors, Scheduling, Big Little Platform.

I. INTRODUCTION

A. Memory Scheduling

It becomes more important to address the energy consumption as the size of data centre increases. High energy consumption cause environmental problems, it also significantly increases the maintenance cost of data centres. The main requirement to save the energy using characteristics of physical memory, the basic approach of reducing memory power consumption is to put the memory nodes that are expected to stay idle for long periods of time into one of power-saving modes.

A large number of virtual machines are joined together in a single computer through virtualization for efficient utilization of computer resources and also for ease of maintenance. It is well known that most of the servers in data centres often stay unutilized or underutilized due to over provisioning [8]. Virtualization allows these underutilized servers to be combined together into a smaller number of physical servers. Moreover, the awaited arrival of new processors with the increased support for virtualization and advances in virtualization software enables more underutilized servers to be joined together. This paper targets these consolidated underutilized servers. To provide sufficient memory for each virtual machine the demand for a large amount of memory has to be increased. This leads to high energy consumption in the memory system.

Understanding the characteristics of physical memory is the fundamental requirement to save memory energy. The entire physical memory is divided into number of memory blocks; each block is the smallest power control unit. This unit is termed as rank in the SDRAM technology; in this paper, this unit is called as a memory node. Each memory node entirely is allowed to be in one of power-saving modes so that it can save energy. Comparing to the normal operating mode, the power-saving mode enables memory nodes to utilize less power without any loss of stored data. To access data in a memory node, its power mode has to be switched to the normal operating mode. In most cases, this process of changing of power mode incurs a substantial delay. Keeping these characteristics of physical memory in mind, the basic approach to reducing memory power braking up, is to put the memory nodes that can be expected to experience comparatively long periods of idleness into one of power-saving modes. A portion of entire physical memory is allocated to each virtual machine in the server consolidation environment, only the memory nodes, which contain the memory pages allocated to it, are accessed when a virtual machine runs. At the same time, other memory nodes are not accessed, and these can be put into one of power-saving modes to reduce memory power dissipation.

Traditional single processor systems use only a single virtual machine to run at a specific time therefore the memory nodes used by the virtual machine are the major source of memory energy consumption. In multi core systems, however, several processor cores simultaneously access the memory since they run a number of different virtual machines at the same time. In these systems, the memory energy consumption varies dynamically, depending on the virtual machines running together. To save memory energy, it is necessary to minimize the number of memory nodes accessed by these virtual machines through a sophisticated scheduling policy.

Scheduling issue in multi core-based computer systems is been mainly focus on, in this paper. Three features are presented, which are essential for reducing energy consumption of memory, and propose power management architecture for memory that supports these features. Based on Memory Power Management architecture, we have come up with several virtual machine scheduling algorithms, goal of which is to minimize the energy consumption in the memory system.

B. Real Time Analysis

Real time performance analysis is critical when the design and integration of embedded software is done to guarantee

DOI10.17148/IJARCCE.2017.6390

that application time constraints will be met at run time. Real-time systems are widely used in the construction of national defence, aerospace, industrial control and in many other fields.

As the majority of general-purpose computers are using Microsoft's Windows or Linux which are open-source as operating system, which are not real- time operating systems, so these systems don’t have enough real-time characters. Presently, a large majority of real-time systems are embedded systems which are built through the embedded processors with embedded operating system.

ENERGY consumption is an important design concern for embedded mobile systems that are battery powered. Displays have been known as the major power consumers in mobile systems. Conventional liquid crystal display (LCD) systems provide very little flexibility for power saving because the LCD panel consumes almost constant power regardless of the display content while the external lighting dominates the system power consumption. Embedded real-time system should be able to respond and deal with system events within the pre-defined limited time. In real-time multi-tasking system, a lot of events and multiple concurrent tasks run at the same time. Therefore, to meet the system response time requirement, we must ensure that each mission can be achieved within the required time frame.

In general, real-time systems are specific application systems, because specific characteristics could ensure their real-time characters to a certain extent. Earlier, real-time systems had no operating system support. To implement multi-task management, engineers must program code for specific practical application [10]. Therefore, these particular software developments have less inheritance for code reuse, maintenance and upgrades which bring a lot of trouble. The emergence of real-time embedded operating system provides a powerful tool for real-time systems’ development and design because of its real-time kernel, multi-task, scheduling and fast interrupt response mechanism and much more. Such real-time characteristics can significantly reduce workload of developers, can cause improvement in development efficiency, and bring a lot more convenience for the maintenance and system up gradation. However, a system that uses real-time operating is not necessarily a real-time system. Real-time operating system is just providing basis for the real-time system, and the most essential elements for a real-time system is to meet the system requirement of task-critical time, which means the system must respond to events in time and complete tasks within the limited time.

II. LITERATURE STUDY

The memory power management architecture is based on the current memory management and scheduling architecture of Xen. This section presents three features which are required to save memory energy based on this architecture

A. Memory Nodes Identification

The first feature is to identify a set of memory nodes which are to be accessed by each virtual machine, which is denoted as M(v) implies the access set of virtual machine v. One a way to get the access set of the virtual machine is Observing memory pages accessed by a virtual machine. In this technique, the page table entries are deliberately deprived; the pages that induce page faults are those used by the virtual machine running at that time. From these accessed memory pages the access set of the virtual machine can be easily obtained. However, this task is necessary to be performed periodically in order to maintain up-to date information on the access sets of virtual machines.

B. Power Mode Transition

The second feature is to control memory power mode by using the identified access sets. The memory power control module handles the power mode of each memory node on the bases of currently used memory nodes [9]. The currently used memory nodes at time t, which are denoted as C(t), indicate set of memory nodes accessed by virtual machines that run on every core processor at time t. Since C(t), is the union of the access sets of those virtual machines, it usually changes when one virtual machine finishes and another begins to run on any of the processor cores.

To reduce memory energy consumption without performance loss, the memory nodes are put in C(t) by the memory power control module, in the standby mode, and other memory nodes in the power-saving mode. Since C(t), may change when the contexts of two virtual machines are switched, the virtual machine scheduler of each processor core informs the memory power control module of the access set of the newly scheduled virtual machine in advance of performing context switch between two virtual machines[9]. After being informed by the virtual machine scheduler, the memory power control module updates the C(t), and transitions the power mode of each memory node if necessary. Transitioning power modes from the power-saving mode to the standby mode incurs a significant delay. If the delay exceeds the time required for a context switch between two virtual machines, the next virtual machine may experience an unexpected delay when accessing memory node whose power mode transition is in progress. In order to avoid this performance degradation, the context switching latency should be larger than the transition delay from the power-saving mode to the standby mode.

C. Memory-Aware Virtual Machine Scheduling

In multi-core systems, the total energy consumption in memory can be reduced further by changing the schedule of the given virtual machines. For example, if the execution sequence of the same virtual machines is changed, the total energy consumption is decreased to 16. Note that the energy reduction stems from the decrease in the number of memory nodes that are put into the standby mode by rearranging the execution order of virtual machines. Therefore, it is necessary to develop a new scheduling policy that reorders the execution sequence of virtual machines in order to save additional memory energy[5]. The fundamental goal of this memory-aware virtual machine scheduling is to minimize the number of memory nodes in the standby mode. In the following sections, we mainly focus on this issue.

Real-time operating system is a subtype of operating system. Lot of its characteristics are similar to common operating system in many respects. It is responsible mainly for the control and management of variety of hardware resources to enable the hardware system to become available, and provide upper level applications with rich system calls. It schedules execution in a timely manner, also manages system resources and provides a consistent foundation for development of application code [5].

A. Components of RTOS

Most of the Real Time Operating System(RTOS) kernels consist of following components:

• Scheduler - The scheduler is present at the heart of every kernel. A scheduler provides the algorithms which are needed to determine which task has to be executed and when.

• Objects- Most commonly the RTOS kernel objects are tasks, semaphores and message queues.

• Services- Most kernels provide services that help developers to create applications for real time embedded systems. These services comprise sets of API calls which can be used to perform operations on objects of kernel or can be used in general, to facilitate the following services:

- Timer Management
- Interrupt Handling
- Device I/O
- Memory Management

Embedded systems are used for many different applications. These applications can be proactive or reactive depending upon the requirements like interface, scalability, connectivity etc.

B. Characteristics

1. Its real time characteristic-Response to events in time and complete tasks within the limited time.
2. The scheduling objective is letting high priority task go first 3. The tasks running on real-time operating system should be certain
4. Some data are highly sharing in real-time operating system

Factors affecting real time characteristics of operating system:

There are varieties of factors impacting a system's real time. Among these factors, operating system and its own factors play crucial roles, including process management, scheduling of task, time of context switching, memory management mechanism, the time of interrupt handle, and so on.

III.CONCLUSION

The selection of right operating system for a specific application has a great impact on performance of real-time system. The improvement of real-time performance of Linux kernel in the embedded application has a far-reaching significance. The complex relationship between the tasks may cause heavy system consumption on internal communication between tasks. The Scheduling and separating the tasks in the beginning will also give us flexibility to apply different algorithms for different task queues. Synchronization mechanism between tasks will decline the real-time performance of system. Lastly, how to use a real-time operating system to implement an actual application system is the key for all embedded system developer.

REFERENCES

[1] Miao Liu et.all, "On Improving Real Time Interrupt Latencies of Hybrid Operating Systems with TwoLevel Hardware Interrupts",IEEE Transactions On Computers, Vol. 60, No 7, July 2011, pp. 978-991.

[2] Mian Dong and Lin Zhong, "Power Modeling and Optimisation for OLED Displays", IEEE Transactions on Computers,Vol. 11, No 9, September 2012, pp. 1587-1599.

[3] L.I. Bing and L.I. Zhong-wen, "Analysis of Linux Realtime Mechanism", Computer Technology and Development, vol. 17(09), Sep. 2007, pp. 41-44.

[4] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T.W.

[1] Keller, "Energy Management for Commercial Servers," Computer, vol. 36, no. 12, pp. 39-48, Dec . 2003.

[5] S. Andrew, Tanenbaum, S. Albert and Woodhull,"Operating Systems Design and Implementation"（Third Edition）, Prentice Hall, January 04,2006.

[6] R. Bianchini and R. Rajamony, "Power and Energy Management for Server Systems," Computer, vol. 37, no. 11, pp. 68-74, Nov.2004.

[7] D. Meisner, B.T. Gold, and T.F. Wenisch, "PowerNap: Eliminating Server Idle Power," Proc. 14th Int'l Conf. Architectural Support for Programming Languages and Operating Systems, pp. 205-216, 200

[8] VMware, "Server Consolidation Overview, Building a Virtual Infrastructure," http://www.vmware.com/solutions/consolidation/consolidate.html,2009.

[9] Prakash Prasad, Dr. Akhilesh Upadhyay , "Reducing Energy for real-time systems through memory scheduling " , ICACSIT 2012,International Conference on Advanced Computer Sciences, Communication and Information Technologies Nagpur, Nov 6, 2012.

[10] Prakash Prasad, Dr. Akhilesh Upadhyay, "Mixed Kernel Architecture for the multiple utility of Operating System", ICACCT 2012, International Conference on advanced computing and communication Technologies, APIIT, Panipat, Haryana, IEEE Computer Society ,Nov 3, 2012.

[11] Prakash Prasad, Dr. Akhilesh Upadhyay, " Advanced Scheduling Policy and its Performance for the Embedded Real time System", IJCAM International Journal of Computer Architecture and Mobility, ISSN 2319-9229 Vol.1 Issue 6, April 2013.

DOI10.17148/IJARCCE.2017.6390