Task-aware Virtual Machine Scheduling for Multi-core Real-time Systems

Yong Park, Dongkun Shin
School of Information and Communication Engineering
Sungkyunkwan University
Suwon, Korea
pangol@skku.edu, dongkun@skku.edu

Abstract

Virtualization allows running multiple operating systems providing the performance isolation. Therefore, it can guarantee the required system utilization to real-time OS when both real-time OS and non real-time OS are serviced by virtual machine monitor (VMM). Virtualization is also an effective technique to manage multiple processors under recent multi-core systems. However, current VMMs cannot support a real-time scheduling in multi-core environment since the real-time task information is unavailable to VMM .

This paper presents a task-aware virtual machine scheduling scheme which assigns the required utilization to each virtual CPU (VCPU) based on the information on the real-time task allocated to the VCPU. Experimental results showed that the proposed scheduling scheme supports real-time systems on multi-core processor.

Keywords: Virtualization, Real-time scheduling, Multi-core.

1. Introduction

Virtualization provides an efficient and isolated duplicate of the real machine called virtual machine (VM) to guest operating systems (OS). Virtual machine monitor (VMM) is the software layer that provides the virtual machine environment [1]. One of advantages of virtualization is the performance isolation which guarantees the required system utilization to guest OS. Therefore, virtualization is useful when both real-time OS and non real-time OS are serviced by VMM. Another advantage is that virtualization can manage multiple processors efficiently under recent multi-core systems. VMM can provide multiple virtual CPUs (VCPUs) to guest OS and manage the utilizations of VCPUs. However, current VMMs cannot support a real-time scheduling in multi-core environment since the real-time task information is unavailable to VMM.

For example, Xen VMM provides a simple earliest-deadline-first (SEDF) real-time scheduler to support real-time guest OS [2]. While the SEDF scheduler is useful at single processor VMs, it cannot support multiprocessor VMs. When a VM has more than one VCPU, guest OS scheduler should allocate a task to one VCPU without the information on the VCPU's utilization allocated by VMM. In addition, SEDF scheduler distributes the total utilization of VM to each VCPU equally. Therefore, the required utilization of a task may not match the utilization of VCPU at which the guest OS assigns the task.

In order to support real-time scheduling in multi-core systems, we propose a task-aware VM scheduling scheme where VMM allocates the utilization of VM based on the task information sent from the guest OS scheduler.

The rest of the paper is organized as follows: Section 2 introduces the related work. In Section 3, we describe the task-aware virtual machine scheduling scheme. Section 4 presents the experimental results. Finally, Section 5 concludes.

2. Related Works

Xen VMM provides the SEDF scheduler for real-time systems, where each VM is specified by a tuple (s, p, x). The slice s and the period p represent the physical CPU utilization to be allocated to the VM. That is, the VM will receive at least s units of time in each period of length p. The boolean flag x indicates whether VM is eligible to receive extra CPU time.

Janus [3] proposed a VMM for real-time tasks. Under Janus VMM, real-time tasks are directly scheduled by the VMM scheduler to guarantee the timing constraint of each task. Therefore, the scheduling policy of guest OS ignored.

Kim *et al*. [4] proposed a task-aware VM scheduler that took into account the I/O bound nature of guest level tasks and correlated incoming events with I/O-bound tasks while making scheduling decisions.

3. Task-aware VM Scheduling

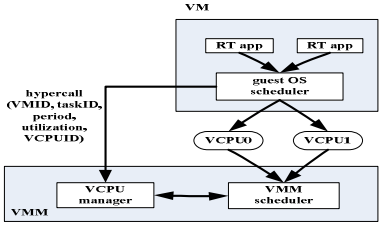

Figure 1: Task-aware VM scheduling architecture

Figure 1 shows the proposed task-aware VM scheduling architecture. The guest OS scheduler sends the task information such as the required utilization and the assigned VCPU to the VCPU manager in VMM. VCPU manager manages the utilization of each VCPU. VMM scheduler is responsible for allocating physical CPUs to VCPUs.

For the task-aware VM scheduling, we modified Xen VMM as follows: When a task is scheduled to a VCPU by the guest OS scheduler, the scheduler determines the required task utilization based on the deadline and execution time of the task. Then, a hyper call from the guest OS to VMM delivers the task information to VCPU manager.

The VCPU manager initializes the utilization of the target VCPU and manages the changes of utilization when VMM scheduler schedules the VCPU at a physical CPU. The VMM scheduler assigns a physical CPU only when the VCPU has a remaining utilization budget. While the current VMM considers only the total utilization of VM, the proposed VMM provides a VCPU-level utilization management in order to support real-time systems.

4. Experiment

To evaluate the task-aware VM scheduler, we modified Xen 4.0.1 and the para-virtualized Linux Kernel 2.6.18. Experiments are performed at a quad core Intel Core2 processor.

Since the processor has four cores, the total available CPU utilization is 400%. We loaded a virtual machine on the modified Xen and assigned 120% utilization to the VM. The VM has two VCPUs, each of which executes one real-time task. We generated five test cases, where each task requires a different CPU utilization as show in Table 1. We observed the deadline miss and the VCPU utilizations under the original SEDF scheduler and the task-aware scheduler.

Table 1: The required CPU utilizations of tasks

Test Case	1	2	3	4	5
Task1	30	40	60	80	90
Task2	90	80	60	40	30

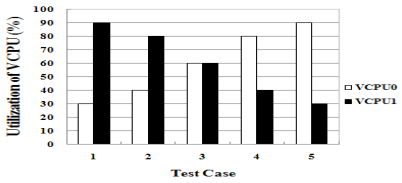

Figure 2: VCPU utilizations under the task-aware scheduler

Under the SEDF scheduler, if the required CPU utilization of a task is higher than 60%, its deadline is missed since the scheduler gives 60% of CPU utilization to each VCPU. However, the task-aware scheduler always meets the deadline of tasks because it can allocate the required utilization to VCPU exactly as shown in Figure 2.

5. Conclusion

In this paper, we addressed the need for VPU-level utilization scheduling under multi-core virtualized systems and proposed a task-aware VM scheduling scheme. By using the task information, the task-aware VMM scheduler can allocate the required CPU utilizations to VCPUs to support real-time tasks. With the modified Xen virtual machine monitor, we showed that the task-aware VM scheduler can guarantee the real-time constraints in multi-core systems.

6. Acknowledgements

This research was supported by KORUS Tech Program funded by the KIAT (KORUSTECH(KT)-2008-DC-AP-FS0-0003).

References

[1] M. Rosenblum, T. Garfinkel, “Virtual Machine Monitors: Current Technology and Future Trends,” *IEEE Computer*, vol. 38, no. 5, pp. 39-47, May 2005.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfiled, “Xen and art of virtualization,” *SOSP*, pp. 164-177, October 2003.

[3] R. Rivas, A. Arefin, K. Nahrstedt, “Janus: A Cross-Layer Soft Real-Time Architecture for Virtualization,” *HPDC*, pp. 676-683, June 2010.

[4] H. Kim, H. Lim, J. Jeong, H. Jo, J. Lee, “Task-aware virtual machine scheduling for I/O performance,” *VEE*, pp. 101-110, March 2009.