International Journal of Sciences
(ISSN 2305-3925)
http://www.ijSciences.com

Research Article
Volume 2, Issue Mar 2013

Exploring CPU Scheduling Strategies in Xen Virtualization Technology

Rajendra Bele[1]✎, Chitra Desai[2]

[1]Dept. of Comp. Sc. Pune University, India
[2]Dept. of MCA, MIT Aurangabad, India

Abstract:

Xen is Open Source more popular virtualization solution provides lot of advantages and ready for various trials due to nature of Open Source. There are lots of migrations in the strategies utilized by Xen .While providing virtualization Xen implies different strategies for Resource Sharing, Memory virtualization, CPU Virtualization, I/O Virtualization etc. In Xen virtualization technology adjusting virtual machine to corresponding physical machine will play important role to enhance performance of the system this is prime duty of CPU Scheduler; hence we have considered CPU Virtualization in detail and attempted to explore CPU scheduling strategies its comparative study as well as research issues related to it.

1. Introduction to Xen:

Xen Originally Started as Research Project at University of Cambridge. It is X86 virtual machine monitor which allows multiple OSes to share same hardware. In the beginning Xen supported only linux server virtual machine, but nowadays it also supports non-modified guest operating system using the hardware supported virtualization technology. There are many companies such as Amazon, IBM and so on, which use Xen for commercial sale. [1] Xen based on the para-virtualization technology. The hypervisor manages physical Resources and provides the services to many guests such as scheduling, memory management, Inter Process Communication and so on. The Dom0 guest domain manages the requests from virtual devices through the frontend driver of guests and applies those requests to the real device. The memory virtualization of Xen uses the direct paging scheme. This scheme allocates all the necessary physical memory when guest starts, and the hypervisor notifies this conversion table to the guest. The guest should manage all the page directories and tables using this table directly. [2]

2. Role of Scheduling in Xen Virtualization

2.1 Role of Hypervisor: In a virtualized environment the hypervisor is important entity between the guests and the hardware any request by the guest to access the hardware is routed through the hypervisor and hence, it is very important for the hypervisor to ensure that all the guests are given fair access to the hardware. Summarily we can say that hypervisor is responsible to provide quality of service to the guests. This quality of service mechanism is provided by scheduling the guests appropriately. A scheduling mechanism guarantee that a guest performs its transactions in its scheduled time and this way, all the guests get a fair chance to access the resources. [3]

2.2 Scheduling in Xen Hypervisor:

CPU Scheduling is critical in server virtualization technology. In order to optimize and allow near native performance scheduling scheme must be efficient and should not waste any processing Cycle. Such schemes are called as work-conserving that is they do not allow CPU to be unused [4]. Xen Hypervisor is responsible for managing the scheduling of guest domain instruction execution with the physical CPUs available in the underlying hardware platform.

✎ **Rajendra Bele (Correspondence)**
✉ belerajendra753@gmail.com

2.3 Scheduling Basics in Xen:
The Xen scheduler acts as a referee between the running domains. It is similar to the Linux scheduler: It can preempt processes as needed, it tries its best to ensure fair allocation, and it ensures that the CPU wastes few cycles. Xen's scheduler schedules domains to run on the physical CPU. These domains, in turn, schedule and run processes from their internal run queues. Because the dom0 is just another domain as far as Xen's concerned, it's subject to the same scheduling algorithm as the domUs. [5]

3. Scheduling Algorithms in Xen:
Xen can use a variety of scheduling algorithms, ranging from the simple to the Complex. An efficient virtual machine scheduling algorithms is important to increase throughput and decrease response time.
Although Xen has shipped with a number of schedulers in the past, Xen developers gone through various algorithms but three algorithms like **BVT, SEDF** and **Credit schedules** have shown considerable performance we're going to concentrate on these three algorithms.

3.1 Borrowed Virtual Time (BVT)

IT is a fair-share scheduler based on the concept of virtual time, is patching the runnable VM with the smallest virtual time first.
BVT provides low-latency support for real-time and interactive applications by allowing latency sensitive clients to "warp" back in virtual time to gain scheduling priority. The client effectively "borrows" virtual time from its future CPU allocation. The scheduler accounts for running time in terms of a minimum charging unit (mcu), typically the frequency of clock interrupts. The scheduler is configured with a context switch allowance C, which is the real time by which the current VM is allowed to advance beyond another runnable VM with equal claim on the CPU (the basic time slice or time quantum of the algorithm). C is typically some multiple of mcu. Each runnable domain to receives a share of CPU in proportion to its weight $weight_i$. To achieve this, the virtual time of the currently running Dom_i is incremented by its running time divided by $weight_i$.[9][10].

3.2 Simple Earliest Deadline First (SEDF)
This algorithm uses real-time algorithms to deliver guarantees. Each domain Dom_i specifies its CPU requirements with a tuple (s_i, p_i, x_i), where the slice s_i and the period p_i together represent the CPU share that Dom_i requests: Dom_i will receive at least si units of time in each period of length p_i. The boolean flag x_i indicates whether Dom_i is eligible to receive extra CPU time (WC-mode). SEDF distributes this slack time fairly manner after all runnable domains receive their CPU share. One can allocate 30% CPU to a domain by assigning (3 ms, 10 ms, 0) or (30 ms, 100 ms, 0). The time granularity in the definition of the
period impacts scheduler fairness.[8][9]

3.3 The credit scheduler: The credit scheduler is a proportional fair share CPU scheduler built from the ground up to be work conserving on SMP (Symmetric Multiprocessing Platforms) hosts. It is now the default scheduler in the xen-unstable trunk. The SEDF and BVT schedulers are still optionally available but the plan of record is for them to be phased out and eventually removed. [8]

Weight and capacity: Each domain (including Host OS) is assigned a **weight** and a **cap**. A domain with a weight of 512 will get twice as much CPU as a domain with a weight of 256 on a contended host. Legal weights range from 1 to 65535 and the default is 256. The capacity (cap) optionally fixes the maximum amount of CPU a domain will be able to consume, even if the host system has idle CPU cycles. The cap is expressed in percentage of one physical CPU: 100 is 1 physical CPU, 50 is half a CPU, 400 is 4 CPUs, etc... The default, 0, means there is no upper cap. SMP load balancing [8]
The credit scheduler automatically load balances guest VCPUs across all available physical CPUs on an SMP host. The administrator does not need to manually pin VCPUs to load balance the system. However, she can restrict which CPUs a particular VCPU may run on using the generic vcpu-pin interface.

Algorithm Description
Each CPU manages a local run queue of runnable VCPUs. This queue is sorted by VCPU priority. A VCPU's priority can be one of two values: **over** or **under** representing whether this VCPU has or hasn't yet exceeded its fair share of CPU resource in the ongoing accounting period. When inserting a VCPU onto a run queue, it is put after all other VCPUs of equal priority to it. [8]
As a VCPU runs, it consumes **credits**. Every so

often, a system-wide accounting thread re computes how many credits each active VM has earned and bumps the credits. Negative credits imply a priority of over. Until a VCPU consumes its allotted credits, it priority is under. On each CPU, at every scheduling decision (when a VCPU blocks, yields, completes its time slice, or is awaken), the next VCPU to run is picked off the head of the run queue. The scheduling decision is the common path of the scheduler and is therefore designed to be light weight and efficient. No accounting takes place in this code path.[8]When a CPU doesn't find a VCPU of priority under on its local run queue, it will look on other CPUs for one. This load balancing guarantees each VM receives its fair share of CPU resources system-wide. Before a CPU goes idle, it will look on other CPUs to find any runnable VCPU. This guarantees that no CPU idles when there is runnable work in the system. [8][9]

4. Conclusion and Future Work

In this paper three algorithms are explored on the basis of available literature and concluded that BVT is preemptive; it is fixed with WC (work conservative) mode only, it displays low overhead in multiprocessor. SEDF is preemptive, performs well in both WC mode and NWC mode but fairness depends upon value of period only.

Credit scheduler is a non-preemptive works fairly in both environments WC (work conservative) and NWC (non-conservative) mode it does automatic load balancing in multiprocessors. This is default scheduler now.

CPU scheduling strategies are crucial for application performance, so different scheduling strategies should be used for different types of application in order to reach the maximum throughput.

Scheduling strategies should be used according to different application workload; hence a self adapting scheduling strategy management is needed.

As a conclusion we can suggest that we should built mechanism in Dom0 to analyze current application workload and change the scheduling strategy of each DomU accordingly might be one of the directions for future works.

References

[1] [1]. Jeanna N. Mathews; Eli M.Dow; Todd Deshane; Wenjin Hu; Running Zen: A Hands on Guide to the Art of Virtualization April, 2008.

[2] [2]. Mohammad Reza Ahmadi, Davood Maleki, “Effect of Virtual Techniques in Data Storage Access”, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops. Pgs 91-96.

[3] [3].Vidya Rao Suryanarayana,”Credit Scheduling and Prefetching in Hypervisors Using Hidden markov Models”, Thesis submitted Dept. of Electrical Engineering and Computer Science Winchita State University.

[4] [4].www.syngress.com

[5] [5].Chris Tekemura and Luke S. Crawford The Book on Xen © 2009

[6] Xiantao Zhang, Yaozu Dong, “Optimizing Xen VMM Based on Intel® Virtualization Technology”IEEE 2008.

[7] [6] Ackaouy,E:The Xen Credit CPU Scheduler,Xen Source:Open Source Enterprise vitrtualization.http://www.xen.org/files/summit_3/Sched.pdf,September 18,(2006).

[8] [7] Analysis and Evaluation of the scheduling Algorithms in Virtual Environment 2009. International Conference on Embedded software and systems.

[9] [8] http://wiki.xensource.com/xenwiki/CreditScheduler

[10] [9].Ludmila Chakasova, Diwakar Gupta and Amin Vahdat. ”Comparision of the three CPU Schedulers in Xen”

[11] ACM SIGMETRICS Performance Evaluation Review, 2007.

[12] [10] Byung Ki Kim_, Jin Kim, Young Woong Ko” E_cient Virtual Machine Scheduling Exploiting VCPU

[13] Characteristics” International Journal of Security and Its Applications V o l. 6 N o. 2, April, 2012.