The Balance Mechanism of Power and Performance in the Virtualization

Jiang Yaoyao

School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China 200240

jyyconan@163.com

Keywords: Virtualization, Energy, Performance.

Abstract. With the datacenter becomes increasingly large in size, the energy consumption in datacenter grows rapidly. Energy saving in datacenter becomes a big theme. In multicore environment, virtualization provides an effective way to address this problem. Virtual machines (VMs) with their own tasks are run on the physical CPUs. Since taking the different task, VMs present different affinity with physical resource. Take advantage of this affinity information is good for VM running performance and energy consumption. In this paper, we introduce three parameters to describe the characteristics of CPU, and design an Affinity Coefficient (AC) computational formula to measure the affinity between virtual CPU (VCPU) and physical CPU (PCPU). According this affinity, we present a novel scheme which combines the VM placement algorithm and frequency scaling to balance the power and performance. This scheme can get better task running EDP in multicore environment.

Introduction

As the datacenter becomes increasingly large in size, the energy consumption in datacenter grows rapidly. The rated power consumptions of servers have increased by more than 10 times over the past ten years [1]. The energy-related costs have become a major economical factor for datacenters. Energy saving is an important topic for datacenters. The most part of power consumption in datacenters comes from computation processing, disk storage, network, and cooling systems. We will study the processing power management throughout the Virtual Machine (VM) placement.

Datacenter virtualization can improve resource utilization, reduce power consumption, simplify management and assure application performance, etc. In cloud computing environments, virtualization technologies are used to run many virtual machines (VMs) on smaller numbers of physical servers. Thus we can significantly reduce the IT costs and energy usage by turning off the unneeded servers. It is important to determine the allocation of VMs to servers such that each VM is allocated with sufficient resources (such as CPU resources, network bandwidth, and memory).

In this paper, we design an Affinity Coefficient (AC) computational formula to measure the affinity between virtual CPU (VCPU) and physical CPU (PCPU). To more reasonable, the formula introduces three parameters to describe the characteristics of CPU. They are CPU utilization, memory bus utilization and the number of network I/O disruption. And calculating AC needn't consider the large difference of each value. And then we present a novel scheme which combines the VM placement algorithm and frequency scaling to balance the power and performance. This scheme can get better task running EDP in multicore environment.

Related Work

The scheduler is an important component of an operating system. It is responsible for deciding which applications or VMs run on the PCPU simultaneously. The scheduling is critical for performance and energy efficiency.

Previous research has investigated the problem of selecting a frequency at which to run a task with given characteristics most efficiently in terms of energy consumption and performance [2]-[4]. Memory-bound tasks can be executed at lower CPU frequencies without significant slowdown, since memory throughput and not CPU speed is the determining factor for their performance. In

contrast, compute-bound tasks run more efficiently at higher frequencies, since lower frequencies prolong their runtime and cause them to consume power for a longer time, often negating the power savings gained by frequency scaling. However, all previous research was based on the assumption that a separate frequency can be chosen for each CPU.

Co-scheduling tasks based on memory bandwidth or other shared resources has been proposed for SMP [5], SMT [6], and CMP systems [7], [8]. To our knowledge, no previous research has addressed resource conscious co-scheduling and frequency selection in combination. However, our experiments with the SPEC CPU 2006 benchmarks suggest that memory contention is becoming more important than cache contention.

A number of approaches have leveraged the dependency between optimal processor speed and memory intensity in the context of heterogeneous multicore processors, that is, chips whose cores support the same instruction set architecture, but operate with different speeds [9]. By assigning tasks to cores based on their memory intensity, runtime, energy, or EDP can be optimized. A related approach assigns a virtual machine monitor to a slower core, since its computations often overlap with I/O [10].

Proposed Approach

The Balance Mechanism of Power and Performance in the Virtualization can be divided into three parts, System Framework, Affinity Monitor and Frequency Scaling.

Part 1: System Framework. The affinity between VM and PCPU is a critical factor for VM placement. To quantify the affinity between VM and PCPU, we present the Affinity Coefficient (AC) to measure it. AC introduce three parameters that can describe the characteristics of VCPUs and PCPUs. They are CPU utilization, memory bus utilization and the number of network I/O disruption. These three parameters correspond to CPU-bound task, Memory-bound and I/O-bound task respectively. To get the better VM placement, the affinity monitor modular collect the information and calculate the affinity between VMs and PCPUs. We use a vector R= (R_1, R_2, R_3) to describe the characteristics of the CPU. R_1 represents CPU utilization, R_2 represents Memory bus utilization, R_3 represents the number of network I/O disruption.

The VCPUs and PCPUs all have their running state features. Since different task demonstrates different characteristics in these three parameters. For easier understanding, the characteristics of PCPU and VCPU are noted by vector R_p= (R_{p1}, R_{p2}, R_{p3}) and R_v= (R_{v1}, R_{v2}, R_{v3}) respectively. We present the Affinity Coefficient between $VCPU_i$ and $PCPU_j$ by $AC_{i,j}$. Affinity monitor modular detect R_{vi} and R_{pj}, and calculate the $AC_{i,j}$. The affinity Coefficient $AC_{i,j}$ can be gotten by Equation (1):

$$AC_{i,j} = 1 - \cos(\theta_{i,j}) = 1 - \frac{\sum_{l=1}^{3} R_{vl} \times R_{pl}}{\sqrt{\sum_{l=1}^{3} R_{vl}^2} \times \sqrt{\sum_{l=1}^{3} R_{pl}^2}} \tag{1}$$

Part 2: Affinity Monitor. VM placement is to match the highest affinity between VCPUs and PCPUs. Frequency scaling is to adjust the most suitable running frequency for PCPU. So the duty of affinity monitor is to get these features information and service for placement and frequency scaling module as follows.

(1) For placement. Affinity monitor detect data R_p and R_v from VCPUs and PCPUs, and calculate all $AC_{i,j}$ and provide them to placement module. Placement module can match the VM and PCPU according these affinity information.

(2) For frequency scaling. Affinity monitor get the PCPUs' features, and it can judge that which kind of task running on the PCPU. It provide such information, and the frequency scaling module could scale the PCPU's running frequency. The frequency scaling detail will be introduced in section 'Frequency Scaling'.

Part 3: Frequency Scaling.

In order to reduce the resource contention, we would gather the VMs with the similar VCPU feature, and place them to the same PCPU. In previous work, we can understand that the Memory-bound tasks can be executed at lower CPU frequencies without significant slowdown and the CPU-bound task will be great influenced by the CPU frequency, and the frequency of CPU is little effect on I/O bound task's performance and energy consumption.

Putting above factors together, we design the formula (2) to estimate the running CPU frequency. The Equation (2) is a linear interpolation equation. The f_{min} is the CPU frequency baseline. According to current running state of CPU and memory bus, Equation (2) could calculate a suitable frequency for PCPU.

$$f(R_{1.cur}, R_{2.cur}) = \frac{R_1(f_{max}-f_{min})(R_{2.max}-R_{2.cur})}{f_{max}(R_{2.max}-R_{2.min})} + f_{min} \tag{2}$$

Where the $R_{1.cur}$ and $R_{2.cur}$ means the value of current CPU utilization and current memory bus utilization respectively, and the f_{max} and f_{min} respectively represent the maximum and minimum CPU frequency. We use $f(R_{1.cur}, R_{2.cur})$ to represent the scaled CPU frequency when the CPU and memory bus state are in $R_{1.cur}$ and $R_{2.cur}$.

Experiment and Performance Evaluation

This section we evaluate the performance of placement algorithm in our experimental platform. Our experiment was implemented in a computer equipped with an I7-3770K core, which has an 8MB L3-Cache. And we use a watts up to measure the computer's power consumption. The computer was virtualized using a kernel-based virtual machine (kvm) implementation, and create several VMs running on hypervisor. To simplify the detection of VCPU's feature, each VM is equipped with one VCPU, this is also the common scenario.

Considering different scenarios, the experiment tested four different kinds of workload which are mixed by different kinds of benchmarks:

(1) Sysbench_cpu*4 gromacs*4 namd*4 provray*4 stream soplex mcf GemsFDTD iperf*4
(2) Sysbench_cpu gromacs namd provray stream*4 soplex*4 mcf*4 GemsFDTD*4 iperf*4
(3) Sysbench_cpu gromacs namd provray stream soplex mcf GemsFDTD iperf*16
(4) Sysbench_cpu*2 gromacs*2 namd*2 provray*2 stream*2 soplex*2 mcf*2 GemsFDTD*2 iperf*8

Two metrics, energy consumption and EDP, were measured respectively for each kind of workload. The results listed below (Fig. 1). Obviously, both the energy consumption and EDP are reduced when implementing our mechanism.
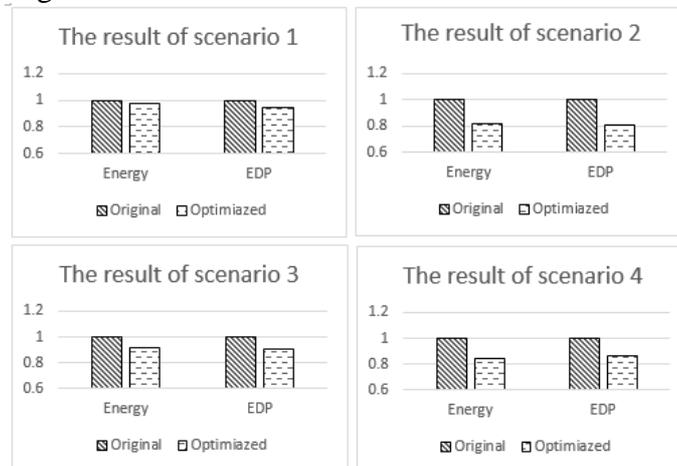

Figure 1. Energy and EDP with different workloads and approaches

Summary

In this paper, we describe the affinity-aware mechanism which can infer the affinity information of tasks in VMs. We introduce three parameters to describe the characteristics of CPU, and design an Affinity Coefficient (AC) computational formula to measure the affinity between virtual CPU (VCPU) and physical CPU (PCPU). According this affinity, we present a novel scheme which combines the VM placement algorithm and frequency scaling to balance the power and performance. From the experiments above, we can find this scheme can get better task running EDP in multicore environment.

References

[1] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No power struggles: Coordinated multi-level power management for the data center," in ACM SIGARCH Computer Architecture News, vol. 36, no. 1. ACM, 2008, pp. 48–59.

[2] V. W. Freeh, D. K. Lowenthal, F. Pan, N. Kappiah, R. Springer, B. L.Rountree, and M. E. Femal, "Analyzing the energy-time trade-off in high-performance computing applications," Parallel and Distributed Systems, IEEE Transactions on, vol. 18, no. 6, pp. 835–848, 2007.

[3] R. Kotla, A. Devgan, S. Ghiasi, T. Keller, and F. Rawson, "Characterizing the impact of different memory-intensity levels," in Workload Characterization, 2004. WWC-7. 2004 IEEE International Workshop on. IEEE, 2004, pp. 3–10.

[4] D. C. Snowdon, E. Le Sueur, S. M. Petters, and G. Heiser, "Koala: A platform for os-level power management," in Proceedings of the 4th ACM European conference on Computer systems. ACM, 2009, pp.289–302.

[5] C. D. Antonopoulos, D. S. Nikolopoulos, and T. S. Papatheodorou, "Scheduling algorithms with bus bandwidth considerations for smps," in Parallel Processing, 2003. Proceedings. 2003 International Conference on. IEEE, 2003, pp. 547–554.

[6] R. L. McGregor, C. D. Antonopoulos, and D. S. Nikolopoulos, "Scheduling algorithms for effective thread pairing on hybrid multiprocessors," in Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International. IEEE, 2005, pp. 28a–28a.

[7] D. Chandra, F. Guo, S. Kim, and Y. Solihin, "Predicting inter-thread cache contention on a chip multi-processor architecture," in High-Performance Computer Architecture, 2005. HPCA-11. 11th International Symposium on. IEEE, 2005, pp. 340–351.

[8] A. Fedorova, M. Seltzer, and M. D. Smith, "Improving performance isolation on chip multiprocessors via an operating system scheduler," in Proceedings of the 16th International Conference on Parallel Architecture and Compilation Techniques. IEEE Computer Society, 2007, pp. 25–38.

[9] S. Ghiasi, T. Keller, and F. Rawson, "Scheduling for heterogeneous processors in server systems," in Proceedings of the 2nd conference on Computing frontiers. ACM, 2005, pp. 199–210.

[10] V. Kumar and A. Fedorova, "Towards better performance per watt in virtual environments on asymmetric single-isa multi-core systems," ACM SIGOPS Operating Systems Review, vol. 43, no. 3, pp. 105–109, 2009.