Journal of Systems Architecture 72 (2017) 42–50

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Precise contention-aware performance prediction on virtualized multicore system

Yuxia Cheng[a,*], Wenzhi Chen[a], Zonghui Wang[a], Yang Xiang[a,b]

[a] Zhejiang University, Zheda Road 38, Xihu District, Hangzhou, China
[b] Deakin University, 221 Burwood Highway, Burwood, VIC, 3125, Australia

ARTICLE INFO

Article history:
Received 11 January 2016
Revised 20 February 2016
Accepted 22 June 2016
Available online 23 June 2016

Keywords:
Multicore
Virtual machine
Resource contention
Performance prediction

ABSTRACT

Multicore systems are widely deployed in both the embedded and the high end computing infrastructures. However, traditional virtualization systems can not effectively isolate shared micro architectural resources among virtual machines (VMs) running on multicore systems. CPU and memory intensive VMs contending for these resources will lead to serious performance interference, which makes virtualization systems less efficient and VM performance less stable. In this paper, we propose a contention-aware performance prediction model on the virtualized multicore systems to quantify the performance degradation of VMs. First, we identify the performance interference factors and design synthetic micro-benchmarks to obtain VM's contention sensitivity and intensity features that are correlated with VM performance degradation. Second, based on the contention features, we build VM performance prediction model using machine learning techniques to quantify the precise levels of performance degradation. The proposed model can be used to optimize VM performance on multicore systems. Our experimental results show that the performance prediction model achieves high accuracy and the mean absolute error is 2.83%.

1. Introduction

System virtualization enables multiple virtual machines (VMs) to share underlying physical machines. Typically, multiple VMs running on the same physical multicore system, which is called VM consolidation, can improve resource utilization. As multicore systems are pervasive from the embedded to the high end computing infrastructures, it becomes increasingly important to exploit performance opportunities and improve the efficiency of virtualized multicore systems.

The performance of bare metal multicore system is growing rapidly. It becomes more and more important to efficiently utilize the multicore resources in the virtualization environment. However, VMs sharing the same physical multicore processor will contend for shared micro architectural resources as Fig. 1 shows. Due to traditional virtualization systems can not effectively isolate shared micro-architectural resources among VMs [1,2], such as shared Last Level Cache (LLC), memory controller, prefetcher and bus bandwidth et al., VMs will encounter different levels of performance degradation [3]. Shared resource contention among VMs not only decreases the overall system efficiency but also hurts the performance stability in each VM [4].

To address these problems, previous researchers proposed cache partitioning [5] and page coloring techniques [6] to prevent contention problems. While these techniques guarantee fairness among different tasks, they lack flexibility and reduce overall cache utilization [7]. Contention-aware scheduling techniques [1,8–10] are proposed to more flexibly address resource contention by co-scheduling cooperative tasks to share resources. Through effectively co-locating tasks to reduce performance bottlenecks, these scheduling techniques can improve system efficiency. But some tasks may still encounter unstable performance [11] due to the inaccurate performance prediction in the scheduling.

To more precisely manage contention problems, researchers proposed performance modeling techniques [12–15] to infer application behaviours when they are sharing physical resources. Typically, based on the performance monitoring events, the model predicts the application's performance behaviours. Then, performance optimization solutions can be applied according to the predictions. However, these modeling techniques were proposed in the non-virtualized environment that mainly focused on the shared cache utilization. Contrary to the non-virtualized environment, the virtualized environment is more complex due to applications running in different VMs are ignorant about each other. The modeling techniques [16–18] previously proposed in the virtualized environment

* Corresponding author.
E-mail address: rainytech@zju.edu.cn (Y. Cheng).

http://dx.doi.org/10.1016/j.sysarc.2016.06.006
1383-7621/

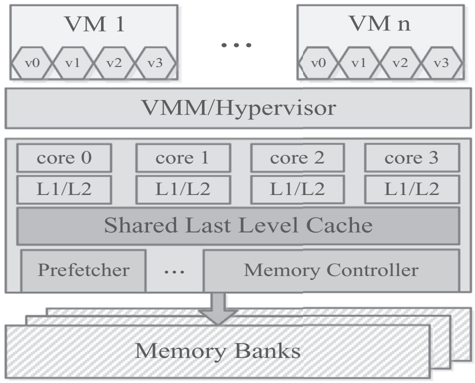

Fig. 1. The simplified architecture of multicore systems.

considered coarse grained factors, such as CPU, memory, storage and network factors. The fine grained micro-architectural level resource contention problems in the virtualized environment needs further investigation.

In this paper, we propose a contention-aware performance prediction model on the virtualized multicore systems. The main contributions of this paper are described as follows:

(1) We investigate the online performance metrics of VMs and identify the performance interference factors that contributes to the VM performance degradation. We find that the data access patterns and working set sizes are the dominant dimensions of resource contention, and then correspondingly design synthetic micro-benchmarks to stress the micro-architectural shared resources. By co-locating micro-benchmarks with real application VMs, we can quantitatively obtain the VMs contention sensitivity and intensity features. These features are correlated to the eventual performance degradation when VMs are co-located on the same multicore system.
(2) Using the obtained contention features, we build the contention-aware VM performance prediction model using machine learning algorithms. We collect the sensitivity and intensity features of applications in VMs, and record the performance degradation results of VMs with different co-location combinations. Using the collected VM's contention features as input variables and the performance degradation as output results, we train the model and build the relationship between contention features and the performance degradation. After the prediction model is built, we can use the model to quantify performance degradation of new VMs with their contention features. The experimental results show that the contention aware prediction model achieves high accuracy and the mean absolute error is 2.83%.

The rest of this paper is organised as follows: Section 2 discusses related work. Section 3 analyses VM runtime performance metrics and contention features. Section 4 presents the performance prediction model. Section 5 shows the experimental results. Section 6 concludes this paper.

2. Related work

In multicore systems, the shared last level cache is the performance critical resource. Previous researchers [12,13,19,20] proposed methods to analyse the shared cache usage in multicore systems. Hardware performance monitoring techniques [21–23] are used to capture program cache behaviours. Tam et al. [24] proposed a technique to estimate shared cache occupation for each core using online cache miss rate curve detection. West et al. [14] proposed hardware performance monitoring based method to predict cache usage for each thread. Based on the cache usage estimation, researchers proposed hardware partitioning [5,25] and page colouring [6,26] techniques to optimize shared cache contention problems in multicore systems.

Tang et al. [11] analysed the performance impact of co-locating large scale datacenter applications due to resource sharing, and showed that there exists both positive and negative impacts of co-locating as application behaviour changes. Mars et al. [27,28] proposed a mechanism named Bubble-up to infer the performance degradation due to co-locating multiple applications on a single multicore server. Zhang et al. [15] and Eyerman and Eeckhout [29] proposed performance models in the multi-thread multicore systems to more precisely co-schedule appropriate tasks. These techniques were proposed in the non-virtualised environment. Contrary to the non-virtualised environment, the virtualised environment has two major differences. One is that there are various kinds of applications running in the virtualised platform, which needs a more comprehensive method to quantify the performance interference. The other is that the applications running in the VMs are ignorant about each other, which needs a smarter scheduler to detect the contention problem.

In virtualised environments, Rao et al. [1] proposed a VCPU migration algorithm to optimize resource contention problems in NUMA (Non-Uniform Memory Access) multicore systems. Liu and Li [2] proposed a NUMA overhead aware hypervisor memory management policy. They introduced a method to estimate the memory zone access overhead using hardware performance counters. Based on the estimation, they proposed two optimization techniques: a NUMA overhead aware buddy allocator and a P2M swap FIFO. Lee and Schwan [30] proposed a region-based scheduling algorithm to manage shared cache resources in multicore platform. Their approach put emphasis on cache/memory-centric scheduling rather than CPU-centric load balancing due to the increasing importance of cache and memory structures to the system performance. However, these techniques only alleviate the interference problems, but can not predict the precise performance degradation due to resource contention.

The VM performance modeling techniques [3,16–18] were proposed to more precisely manage resources in the system. Govindan et al. [3] proposed a method of estimating the cache usage of applications through active probing with the synthetic cache loader benchmark, and uses the cache usage estimation to predict performance degradation of applications upon consolidation with other applications. Kundu et al. [16,17] proposed a method of modeling the performance of VM-hosted applications as a function of resources allocated to the VM and the I/O resource contention it experiences. Chiang and Huang [18] proposed the I/O interference aware scheduling for data-intensive applications in virtualized environment. They used the non-linear models to capture the bursty I/O patterns in data-intensive applications and incorporated the model into the VM scheduling systems. Unlike these modeling techniques, our proposed contention-aware prediction model considers more comprehensive and fine grained micro architectural resources of multicore virtualized systems. In our proposed method, the application contention features only needs to be collected once, then the model can easily utilise these features to quantify the performance degradation of various application combinations with machine learning algorithms.

3. VM runtime analysis

In this section, we describe the method of collecting VM runtime performance metrics and analyse resource contention factors that contribute to VM performance degradation.