Future Generation Computer Systems 29 (2013) 1871–1884

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

# Black box scheduling for resource intensive virtual machine workloads with interference models

Sam Verboven *, Kurt Vanmechelen, Jan Broeckhove

*Department of Mathematics and Computer Science, University of Antwerp, Middelheimlaan 1, 2020 Antwerp, Belgium*

## HIGHLIGHTS

- The predictive capacity of several system-level metrics is evaluated.
- Different models are used to predict slowdown for multiplexed workloads.
- A novel approach using SVM-based classification and prediction is suggested.
- Practical benefits are demonstrated using prediction-based scheduling algorithms.

## ARTICLE INFO

*Article history:*
Received 12 July 2012
Received in revised form
22 April 2013
Accepted 29 April 2013
Available online 9 May 2013

*Keywords:*
Virtualization
Xen
Profiling
Performance modeling
Support vector machines
Scheduling

## ABSTRACT

Modern datacenters consist of increasingly powerful hardware. Achieving high levels of utilization on this hardware often requires the execution of multiple concurrent workloads. Virtualization has emerged as an efficient means to isolate workloads by partitioning large physical resources using self-contained virtual machine images. Despite the many advantages, some challenges regarding performance isolation still need to be addressed. Unmanaged multiplexing of resource intensive workloads has the potential to cause unexpected variances in workload performance.

In this paper, we address this issue using performance models based on the runtime characteristics of virtualized workloads. A set of resource intensive workloads is benchmarked with increasing degrees of multiplexing. Resource usage profiles are constructed using the metrics made available by the Xen hypervisor. Based on these profiles, performance degradation is predicted using several existing modeling techniques. In addition, we propose a novel approach using both the classification and regression capabilities of support vector machines. Application clustering is used to identify several application types with distinct performance profiles. Finally, we evaluate the developed performance models by introducing several new scheduling techniques. We demonstrate that the integration of these models in the scheduling logic can significantly improve the overall performance of multiplexed workloads.

© 2013 Elsevier B.V. All rights reserved.

## 1. Introduction

Virtualization has become a widespread technology used to abstract, combine or divide computing resources in order to allow resource requests to be described and fulfilled with minimal dependence on the underlying physical hardware. Using virtualization, an application and its execution environment can be managed as a single entity, a *virtual machine* (VM) [1], of which the configuration can be captured in a single file, a virtual machine *image*. These virtual machine images can be deployed in a hardware-agnostic manner on any hardware that hosts a compatible *hypervisor* or *Virtual Machine Monitor*. The hypervisor is a software component that hosts virtual machines, also referred to as *guests*. This software layer abstracts the physical resources from the virtual machines by providing a virtual processor and other virtualized versions of system devices such as I/O devices, storage, memory, etc., [2,3]. Hypervisors thereby offer flexibility in partitioning the underlying hardware and ensure some degree of isolation between the different virtual machines sharing these resources.

Although the hypervisor provides adequate isolation on many levels (e.g. security, faults, …) *performance interference* can still be an issue, particularly with resource intensive workloads [4]. Each virtual machine is allocated a subset of the available resources and requires the hypervisor's cooperation to complete certain tasks (e.g. disk or network I/O). When multiple VMs share the same hardware, bottlenecks can occur both on the hardware as well as the hypervisor level. This is partially a consequence of the isolation between the hypervisor and guests, causing multiple guest schedulers to work independently without knowledge of each other. When compared to a single operating system running

* Corresponding author. Tel.: +32 3 265 34 85; fax: +32 3 265 32 04.
*E-mail address:* sam.verboven@ua.ac.be (S. Verboven).

0167-739X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.04.027

comparable workloads this can cause sub-optimal results [5]. The evolution toward an ever increasing amount of CPU cores per server and relatively slower gain in I/O performance poses additional challenges. Resource contention problems will likely become even more important in the future as more VMs share the same hardware in an effort to increase utilization. Improvements in VM scheduling and the reduction of virtualization overhead can partially mitigate these issues. In the context of datacenters where VMs can easily be migrated between hosts, the problem can be addressed at a higher level through intelligent scheduling. This approach requires more insight into the resource consumption of individual workloads and their impact on other workloads. Using this information, a more informed scheduling decision can be made which reduces the impact of resource contention.

In this paper, we build an interference model providing the information needed to automate such scheduling decisions. Using a combination of tools, accurate metrics are obtained from a Xen-based [6] hypervisor hosting multiple VMs on a multi-core system. A combination of CPU usage information, cache hit/miss rates and various I/O metrics is obtained for each VM. A set of diverse applications is benchmarked in two different setups. First, we benchmark all applications in a linear scaling scenario where slowdown is calculated when multiplexing VMs with identical workloads. Interference will likely be highest when concurrent applications use similar resources. Therefore, performance prediction based on the linear scaling data can provide an upper bound on the performance degradation due to interference. Second, we benchmark all applications in a pairwise manner to obtain resource contention information for mixed workloads. To provide reference points for less resource intensive workloads, a number of benchmarks were modified to require fewer resources. In order to reduce the required application set and model complexity, network usage was not yet taken into account and VMs were assigned a single virtual CPU. An approach commonly used in related work [7–9].

We evaluate several techniques to predict performance and classify applications. A novel approach using both the classification and regression capabilities of support vector machines is presented and evaluated. Using *k*-means clustering, we automatically define application types which have distinct behavior under contention. The scheduling potential of performance predictions and application types is evaluated using three new scheduling techniques. The scheduling problem tackled in this contribution comprises the placement of a static set of applications known a priori on a fixed number of servers. Additional complexity in terms of scheduling methodology, such as online arriving applications or admission control, is not taken into account and is left for future work. Finally, we compare a coarse-grained scheduling technique based on application types to a more fine-grained prediction-based approach. We demonstrate that the use of performance predictions in the scheduling logic can significantly increase the overall performance of multiplexed workloads.

Previous work has provided important insights and advances regarding performance modeling and application classification in virtualized environments. Nonetheless, several key limitations can be found in the most closely related efforts. First, a relatively small amount of application profiles is used to validate prediction techniques [8,10–19]. Most use between one and five applications, potentially augmented by using different configuration options. This is a consequence of the significant effort required to obtain a more diverse training set. Second, application sets are restricted to only include benchmark applications limited by resource availability [7,8,16,20]. Our experiments show that models trained using only these application types are not reliable when used to predict the performance of applications with less strenuous resource requirements. Third, models are trained and evaluated with a limited amount of concurrent workloads [7,8,14,17,19]. Benchmarking application sets using more than two VMs results in an exponential increase in possible configurations. However, with the increasing amount of CPU cores in modern servers this restriction can reduce the relevance for practical applications. Fourth, models are often used to predict and optimize application performance by dynamically reconfiguring the resources allocated to a VM [8,14–19], whereas our approach models performance interference using fixed VM resource allocations. Instead of allocating additional resources, we reduce interference using high-level VM scheduling. Finally, to the best of our knowledge, the combination of classification and performance interference modeling has never been analyzed by developing and evaluating a black box VM scheduling algorithm. In this paper we address all these elements in a single body of work. A broader and more detailed comparison with related efforts can be found in Section 2.

Our contributions can be summarized as follows.

- The predictive capacity of several system-level metrics is evaluated with regard to the slowdown experienced by multiplexed workloads. Experiments evaluate up to nine concurrent VMs on an octa-core server.
- An extensive and diverse set of representative virtualized workloads is profiled. An uncommon feature is the evaluation of workloads not limited by resource constraints.
- Both previously tested and new non-linear modeling approaches are compared on multiple training sets.
- A novel approach using support vector machines (SVM) for both classification and regression analysis is suggested to solve the limitations encountered with the existing modeling solutions.
- Practical applications are demonstrated by implementing and evaluating several prediction-based scheduling algorithms. The value of adding fine-grained application profiles is demonstrated.

The remainder of this paper is organized as follows: Section 2 discusses the existing work and recent developments in the field. An overview of the performance modeling approach is described in Section 3. Section 4 details the challenges involved in obtaining accurate metrics. In Section 5 we describe the benchmarks and frameworks used to gather our performance results, and our experimental setup. Several modeling techniques are explored in Section 6, followed by an introduction on application clustering in Section 7. Integration of the performance models in scheduling algorithms is discussed in Section 8. Our final conclusions can be found in Section 9. Section 10 concludes the paper with a discussion on future work.

## 2. Related work

One of the efforts closely related to our current approach can be found in [7]. Koh et al. consider the impact of performance interference on competing applications in a virtual environment. They collect various system-level workload metrics and perform both application clustering as well as slowdown prediction using two concurrent VMs. Our approach differs in three major aspects. First, their experimentation was limited to two VMs on a dual-processor machine. Second, their regression analysis was limited to basic linear regression, which they show to be less accurate than weighted means. We have extended our modeling to non-linear regression efforts by using support vector machines. Finally, we also investigate how scheduling decisions can be improved through the application of performance interference models.

Hoste et al. [21] also used program similarity to predict the performance of profiled applications. However, their focus was on the evaluation of new processor architectures instead of performance interference. Predictions were made by taking a weighted average

Download English Version:

https://daneshyari.com/en/article/424611

Download Persian Version:

https://daneshyari.com/article/424611

Daneshyari.com