Computers and Electrical Engineering 000 (2017) 1–19

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

Analysis of parallelized libraries and interference effects in concurrent environments[☆]

Fabio Licht[a,*], Bruno Schulze[b], Luis C.E. Bona[c], Antonio R. Mury[a]

[a] *Catholic University of Petropolis (UCP), Petropolis, RJ, Brazil*
[b] *National Laboratory for Scientific Computing (LNCC), Petropolis, RJ, Brazil*
[c] *Department of Informatics, Federal University of Parana (UFPR), Curitiba, Brazil*

A R T I C L E I N F O

Article history:
Received 31 March 2017
Revised 25 August 2017
Accepted 28 August 2017
Available online xxx

Keywords:
System performance
Affinity
Application classes
Performance interference

A B S T R A C T

Cloud computing is being presented as the technology of the moment, and an additional resource for High-Performance Parallel and Distributed Computing, particularly regarding its use in support to scientific applications. This paper evaluates the concept of affinity regarding the combined effects of the virtualization layer, applications classes and parallel libraries used in the applications hosted in cloud computing. Affinity or interference are here presented as the degree of influence that one application has on other applications when running concurrently in virtualized environments hosted on the same real environment. The results presented here show how parallel libraries used in application implementation have a significant influence and how the combinations of these types of libraries and classes of applications could significantly influence the performance of the environment.

1. Introduction

Growing concern about the quality of services provided by cloud computing has led researchers to seek mechanisms and methodologies to analysis and assist scheduling and the allocation of applications on computational resources. In this sense, to know how those applications interact when hosted in virtualized environments and the effects caused by concurrency on real resources can contribute to minimizing performance losses and instability among those environments. In a consolidated virtualized environment or in a cloud environment, in a single physical resource there are multiple virtualized environments where just one process, running in one of these virtualized environments, could cause the degradation of the whole physical resource and consequently could impact on all other existing environments on the same host.

Moreover, the specific case of cloud computing and its use as a support to high-performance parallel and distributed computing, especially from the perspective of scientific applications, studies are devoted to ascertaining the effect of the virtualization layer on the performance of those types of applications with different requirements in comparison to business support applications. The studies performed usually evaluate the loss of performance between clouds and do not address the effects of concurrency among different virtualized environments hosted on the same real cloud resource [1–4].

Since clouds can be shared and host several services, some machines may be hosting several virtualized environments. These virtualized environments in the same physical resource, share: memory, disk and mainly processor and this is the

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. S. A. Aljawarneh.
* Corresponding author.

http://dx.doi.org/10.1016/j.compeleceng.2017.08.028
0045-7906/© 2017 Elsevier Ltd. All rights reserved.

main motivation for this study, in which we intend to analysis which types of applications can coexist in these physical machines, even if executed in different virtualized resources. In this sense, it is worth mentioning that there are already classes of applications that allow analyzing the behavior of an existing type of application in an actual or virtualized machine, but the analysis of competition among these types of applications in shared environments is where this study is inserted.

The present work is part of research that has generated a Ph.D. thesis [5] and a book chapter [6]. In this paper, we propose the concept of affinity, defined here as the effect of an application over another, when there is the need to share the same physical resource by several virtualized environments. The closest concept to that proposed in this paper was presented by Ravi et al. [7], whose framework allows applications to run transparently inside virtual machines in one or more CPU's, seeking efficient virtualization of these CPUs and providing an indication of potential performance improvements in the consolidation process of the kernel.

In addition to [5], other studies present the use of Dwarfs to categorize and analyze the behavior of applications without competitive environments, but in no other work was analyzed competition among applications in a shared cloud environment and particularly the kind of parallel library used in the implementation of the algorithms.

We considered in this work that because there is a behavioral history of some types of applications in free competition, this behavior would be maintained, even when the applications are executed in cloud environments and that there is interference when we execute more than one application in an environment. And what explains this is the sharing of processing in shared clouds. It remains to an analysis of this interference can be minimized with the right choice of types of applications that can coexist in the same resource.

We hope that with the result of this research it could be possible to create scheduling algorithms for the use in shared cloud computing environments. This survey has been possible given the thousands of tests performed in shared environments and taken their times with and without concurrency.

To be able to validate the concept of affinity presented in this work, we used the application class approach. There are already studies that seek to categorize applications into classes, grouping these applications based on their characteristics in terms of use of computational resources. This type of categorization allows newly developed applications to be associated with one of these classes, bringing benefits by raising the possibility of being able to predict the behavior of these applications when running in a virtualized environment. In the tests, we used the Dwarf approach [8]. From the thirteen Dwarf classes proposed, we selected three classes of applications and four algorithms for this study. In one of three classes, we selected two algorithms to be used in order to verify different domains within the same class. The selected classes represent the following types of applications: Dense Linear Algebra, Structured Grid, and Graph Traversal.

For the verification of the affinity level among classes, the tests analyzed the effects that the use of concurrently virtualized environments, competing for the same real resources, would have on the performance of these classes of applications combined with different types of libraries, aiming to determine which combinations could be consolidated into the same real resource and which combinations should be avoided. For the evaluation of the results, two types of analysis were conducted to examine the average performance loss (percent loss) and the distance between these losses (stability of the environment).

The paper is organized as follows: Section 2 presents the literature review. Section 3 presents the methodology and the testing environment. Section 4 presents the heterogeneous performance evaluation assessment. Section 5 presents the results obtained and Section 6 presents the conclusions.

2. Literature review

With the goal of categorizing the styles of computing used in scientific computing, the work of [8] has identified seven numerical methods that he believed were important to science and engineering. Colella has introduced the "Seven Dwarfs" of scientific computing. Kaltofen [9] defines Dwarf as follows: "A dwarf is an algorithmic method that captures a pattern of computation and communication". This pattern is important in the proposed work to provide a basis for knowledge and characterization of types of applications so that when we assess that a certain application can be run in shared environments alongside other applications without degrading the shared resource, one deduces that other applications of the same type (dwarfs) may also compete for the same feature without degrading the resource. The Berkeley team increased dwarfs classes to thirteen, attempting to cover a larger number of standards in scientific applications [10].

In this work, the dwarf classes being used are Dense Linear Algebra (DLA), Structured Grid (SG) and Graph Traversal (GT) [11]. These three classes were chosen because they cover a great number of scientific applications in various scientific fields, as shown in Fig. 1. In Figure, you can see the scope of the chosen algorithms. The Dwarfs types are shown in the lines and the application domains in the columns. It is possible to verify that the three selected classes cover to a greater or less degree all the domains. Understand, to a greater or lesser extent, the intensity of the color. The darker the more comprehensive, the lighter the less comprehensive. Highlighting that the Vacation Reservation System domain is only covered in the Dwarfs classes by Graph Traversal. That was why it was chosen.

The benchmark used for testing was [12], a suite for heterogeneous computing which helps in the analysis of hardware platforms, particularly GPU's (Graphics Processing Units) and CPU's (Central Processing Units). The applications in the package are based on the Dwarf classifications [13]. This work tested OpenCL, or OCL, a parallel programming library that allows us to obtain a standard for writing applications that access all available programming resources, both on CPU's and GPU's as well as for other processors. OpenMP, or OMP, is an application programming interface (API) that is supported by most