Doctoral Thesis Report

Thesis: Reproducible Partial-Load Experiments in Workload Colocation Analysis

Author: Andrej Podzimek

Reviewer: Tomás F. Pena

Submitted: August 2016

The thesis is situated in the context of performance analysis and optimization on multi-chip and multi-core SMP systems. More specifically, it tries to cover an aspect that is not well analyzed in the modern literature: study the performance behavior of applications under partial processor loads. This problem is well worth investigating, since most of the time, systems in data centers are running under this conditions.

The thesis offer two main contributions to the problem. The first and most important is the design and implementation of Showstopper, a tool that allows the developers to determine the dependency of application throughput on partial processor load. Showstopper is a load control tool based on feedback control, which generates partial processor loads, using existing benchmarks, but controlling them so the CPU load can be sustained to a required value or even follow an user specified load trace. Showstopper is workload- and platform-agnostic and have a modular design, which makes it easy to maintain and modify and reduces the memory footprint, since modules are loaded on demand.

The use of Showstopper to analyze the effects of processor sharing on multi-chip and multi-core SMP systems is the thesis' second main contribution. The load control provided by Showstopper is used to infer the relationship between benchmark throughput and a range of partial loads, showing that this relationship is often platform-dependent, non-linear, and even non-monotonic. The results show the impact of workload type, CPU load, and CPU pinning in the overall performance, due to the interference betweens pairs of computationally-intensive workloads. Of special interest is the study of the impact of CPU pinning on the performance interference between the two (background and foreground) workloads. The experiments show how the background load determines the best possible CPU pinning configuration. KVM and LXC containers were used as virtualized environments.

The thesis is presented as a compendia of five papers. The first paper, corresponding to chapter 5, is not directly related with the thesis, but, as is said in the introduction it is "a study that sparked the idea of reproducible partial-load experiments". However, this paper was recently accepted for publication in IEEE TNSM (it is not even published right now), so it is difficult to see how it could inspire the idea of Showstopper. Maybe, the paper from the DNS 2013 conference should have been included instead of the previous one.

The following two papers (MASCOTS 2013 and MASCOTS 2014) introduce the design of Showstopper and give an approximation to the CPU pinning problem. In the CCGRID'15 paper, a thorough set of experiments using KVM virtual machines and LXC containers is presented. The experiments cover different combinations of workloads and

Centro Singular de Investigación en Tecnoloxás da Información, CiTIUS
Universidade de Santiago de Compostela
Rúa de Jenaro de la Fuente Domínguez, 15782 - Santiago de Compostela, SPAIN

virtualized environment types so as different CPU pinning configurations. It was worth remarking that this paper was honored with the Best Paper award in the CCGRID'15 conference.

The last paper was published in FGCS, a very prestigious journal. It summarizes the entire thesis, extending the experiments presented in CCGRID'15 to include a study on performance impact of virtualization (KVM) in comparison to resource containers (LXC).

All the presented results have been actually refereed by several reviewers, which is guaranteeing the interest, validity, and quality of the thesis. I only have a few comments and questions

- The experiments are performed on NUMA systems. However, the influence of memory access is not considered. An important factor influencing performance is the latency accessing to main memory in programs for which data does not fit in cache. It would be interesting to analyze how the memory bus contention influences the performance, additionally to the CPU load.
- One of the main conclusions of the thesis is that processor pinning configuration should no longer be a static design attribute based on rules of thumb and that large-scale deployments can benefit from dynamic pinning adjustments based on a careful analysis of performance interference and power consumption characteristics of workloads. The question is, can Showstopper help with this issue? How could it be extended so to dynamically set the best pinning configuration based on the actual workload?
- In page 18, when describing the space module, it is said that this module controls how the required global duty cycle is distributed among the controlled processes, making no assumptions about the kernel's scheduling decisions, because "There are means to force a kernel to spread load across processors in a particular way, but none of them expose a standard portable interface". I am not sure about that. In fact, in Linux, libnuma and numactl allows to runs processes with a specific NUMA scheduling or memory placement policy, and they are quite standard.
- The selected benchmarks, DaCapo and ScalaBench, both executed on the Java Virtual Machine. It would be interested to see how Showstopper behaves with other kind of benchmarks, as Parsec, to eliminate the influence of the JVM. All the experiments were executed with OpenJDK 1.7 JVM, would be the result any different using other JVM, e.g. Oracle Java 8?

As a final conclusion, **I think that the thesis clearly proves the authors ability for creative scientific work and meets the requirements for conducting the public defense.**

Santiago de Compostela, August 23rd, 2016

Tomás Fernández Pena, PhD

Centro Singular de Investigación en Tecnoloxás da Información, CiTIUS
Universidade de Santiago de Compostela
Rúa de Jenaro de la Fuente Domínguez, 15782 - Santiago de Compostela, SPAIN