



Thesis Overview:

Scheduling Elastic Machine Learning Process through Containers Coplanificación de procesos maleables de aprendizaje automático mediante contenedores

Leandro Ariel Libutti 

III-LIDI, National University of La Plata, Argentina.

Master in High Performance Computing

Thesis Advisors: Dr. Francisco Igual  and Dra. Laura De Giusti 

{llibutti,ldgiusti}@lidi.info.unlp.edu.ar, figual@ucm.es

Motivation

In the era of new-generation multi-core and many-core servers, some of which boast hundreds of general-purpose cores, fully utilizing such resources with a single application has become increasingly challenging. In response to this issue, containerized applications are often co-scheduled for execution, following a strategy known as *horizontal scalability*. In environments where multiples applications are executed concurrently, and where the arrival and lifetimes of these applications exhibit heterogeneity, resources are dynamically released or requested. While there exist mechanisms enabling the reconfiguration of resources allocated to a container during runtime, the applications operating within these containers often lack the necessary adaptability to accommodate such dynamic elasticity. This absence of elasticity support within applications frequently results in well-known scenarios of *oversubscription* when the allocation of resources to a container is reduced or *underutilization* when it is increased. The rigidity in resource assignment to applications/containers is not itself a problematic feature. Static resource assignment offers benefits like isolation, resource contention control, and determinism, but it becomes problematic in environments with multiple heterogeneous applications running concurrently.

Objectives

The following objectives were established for this thesis:

- Modification of the resource management scheme within an application/framework to allow dynamic selection of the parallelism of operations (elasticity).
- Design and implementation of an internal controller for each container to manage dynamically allocated computational resources, and a communication mechanism between the system and the application confined in the container.
- Design and implementation of a scheduler for containers running elastic applications using orchestration techniques to efficiently manage the computational resources of the system.

Contributions

The contributions of the thesis are summarized as follows:

1. Underutilization and Oversubscription in containers execution

In this contribution, we provide evidence that demonstrate that *oversubscription* or *underutilization* effects appear in scenarios in which the amount of resources assigned to a container is decreased/increased while the confined application is not requested to adapt to these modifications accordingly

2. Addition of elasticity in specific machine learning framework (Tensorflow)

We describe the necessary changes in a specific application (Tensorflow) to accommodate on-demand resource elasticity for a specific resource type (number of cores used by the application). The CPU resources are managed through the Eigen library, that adapts to allow increasing and decreasing the number of active threads in execution.

3. Design and implementation of ad-hoc scheduling engine for ML containers

We propose an ad-hoc scheduling engine equipped with elasticity support, in which containerized applications are scheduled for execution with support of dynamic resource assignment or re-assignment at the initial execution point or during the lifetime of the container. This scheduler is equipped with mechanisms not only to modify the resource assigned to a container, but to accordingly modify the resource usage of applications within them.

4. Advantages of the use of elastic scheduling in ML containers

We give evidence of the potential in terms of productivity, time-to-completion and resource usage of the different policies for a coupled container-application resource management compared with a static assignation of resources for a modern multi-core architecture and different workload profiles.

Conclusions

This thesis presents a coupled elasticity-aware scheduling methodology for applications encapsulated within containers. This methodology entails resource management at two distinct levels: firstly, at the container level, utilizing Docker's resource allocation and reallocation mechanisms, particularly concerning processor cores; and secondly, at the application level, where Tensorflow serves as an illustrative application, showcasing the development of a pliable, adaptable application. Our empirical performance assessments underscore the imperative nature of this two-tier elasticity strategy in addressing the challenges associated with *oversubscription* and resource *underutilization* at the application level. Concurrently, it ensures an efficient utilization of processing cores within modern multi-core server environments, wherein the proliferation of compute units is on the rise, and any inadequacies in their management have a direct impact on performance and energy efficiency, thereby incurring costs.

Future Works

The development of this thesis opens the opportunity for new lines of work such as the following:

- Add support for heterogeneous systems in the scheduler. Integrate the management of accelerator resources such as GPUs and edge boards (TPUs).
- Evaluate the energy efficiency of the Elastic Application Scheduler and compare it with other schedulers in the market such as Kubernetes, Marathon, Cloudify, among others.
- Analyze the use of features developed by GPU vendors, such as Multi-Stream, Multi-Process Service (MPS), Multi-Instance GPU (MIG) and virtual GPUs (vCS) for runtime scheduling and resource management support.
- Explore resource management in other ML frameworks such as PyTorch and Caffe, in order to extend elasticity to more applications and perform comparative studies between them.
- Explore resource management in others HPC applications.

Publications with this Thesis Work

- 1) Leandro Libutti, Francisco D. Igual, Luis Pinuel, Laura De Giusti, Marcelo Naiouf. Benchmarking performance and power of USB accelerators for inference with MLPerf. 2nd Workshop Accelerated Mach. Learn.(AccML), pp. 1-15. 2020.
- 2) Leandro Libutti, Francisco D. Igual, Luis Pinuel, Laura De Giusti, Marcelo Naiouf. Towards a Malleable Tensorflow Implementation. In: Rucci, E., Naiouf, M., Chichizola, F., De Giusti, L. (eds) Cloud Computing, Big Data & Emerging Topics. JCC-BD&ET 2020. Communications in Computer and Information Science, vol 1291. Springer, Cham. https://doi.org/10.1007/978-3-030-61218-4_3

Citation: L. A. Libutti. "Thesis Overview: Scheduling Elastic Machine Learning Process through Containers". Journal of Computer Science & Technology, vol. 23, no. 2, pp. 190-191, 2023.

DOI: 10.24215/16666038.23.e17

Copyright: This article is distributed under the terms of the Creative Commons License CC-BY-NC.