ICAI 2023
12th International Conference on Applied Informatics
Eger, Hungary, 2023

Three Level Benchmarking of Singularity Containers for Scientific Calculations

Péter Polgár[a], Tamás Menyhárt[a], Csanád Bátor Baksay[a], Gergely Kocsis[a], Tibor Gábor Tajti[a], Zoltán Gál[a]*

[a]University of Debrecen, Faculty of Informatics
kocsis.gergely@inf.unideb.hu, tajti.tibor@inf.unideb.hu

Abstract

In this work we present our results of benchmarking Singularity containered versus native software environment runs of scientific calculations on three different levels of complexity. During our investigations we have applied up-to-date hardware and the latest available software environments of the beginning of 2023. To get a more detailed picture we applied three different aspects during our work, namely we ran separate runs of CPU, RAM, and I/O calculations on micro- and mid-level while as the top le vel we ran one big pipeline of different calculations being intensive from all the previous aspects. As a result of our investigations we have shown that still and again Singularity containered runs of scientific calculations provide almost similar (or in some cases even better) performance indicators as the same runs on native software environments.

1. Introduction

The tools and methods of scientific computing undergo an endless progress providing the possibilities for researchers to achieve more and more complex or accurate results by using better and better facilities. In the mean time, however, history has shown that following the new trends of tools and methods almost naturally leads to compatibility issues.On the other hand it is also not rare that the same

*This work has been supported by the project TKP2021-NKTA of the University of Debrecen. Project no. TKP2021-NKTA-34 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the TKP2021-NKTA funding scheme.

calculation or simulation has to be ran in different environments (various hardware and/or software setups). Today's answer to provide a solution to these issues is the use of containers [3]. By the use of containers researchers can build their uniform running environments inside a separated area and run their calculations or simulations on them without any regard on the details of the real host's hardware and software setup. From this aspect containers work like virtual machines however as most findings has shown they use much less resources making them more efficient. In the case of scientific calculations Singularity container (or its upgraded fork, Apptainer) is most often used to beat the incompatibility issues between different environments [1]. The question, however, always (and always again) arises if the application of these containers affects the performance of the calculations and if yes, how?

2. Problem Formulation and Applied Methodology

Since the arise of containerization technology several investigations have been conducted to describe the pros and cons of the use of it. While in most of the cases these investigations have shown that the use of containers do not affect the performance much (in some cases they can even beat the performance measures of bare metal environments) [2, 3], it is also clear to see that the details change as new versions of the underlying hardware and software systems are presented. For this reason in this work we compare scientific calculations of three levels of complexity on bare metal and in containerized environments from three different aspects (CPU, RAM, I/O) to find out if the actual setups (in early 2023) has changed the details of these results or not.

As the first level we do „micro-benchmarking" by running huge amounts of simple calculations repeatedly. On this level we can easily find the exact roots of performance weaknesses or strengths of Singularity containers (or native environments). The other benefit of this level is that these tests are really easy to be scaled so we can study the size dependency of the time overheads making it more easy to find out if they are constants or they have some linear or non-linear depencency on the size of the jobs.

The second level of benchmarking is the use of well-known benchmark tools [3, 4]. This level provides us the possibility to compare our results to others'. In the current state of our work we focus on CPU, RAM and I/O intensive calculations and tools since results about these are easy to be compared to the results of the previous level of our investigations.

As the topmost level of ours we run an existing pipeline of scientific calculations related to social sciences. Naturally the exact results of these calculations are not important for us now, but the run times and other performance indicators provide us an understanding about how far the behaviour of real computation scenarios are from the cases of micro-benchmarking and the well known benchmarking tools.

3. Results and Conclusions

As the results of our studies we have shown again that the use of Singularity containers for providing uniformized environments to scientific calculations does not affect the run times of the calculations in an unaffordably negative way. Even in some cases the use of containers can beat the native sotware environments on all levels of our investigations with the most up-to date tools at the beginning of 2023.

Table 1. Results of comparing native and containered benchmarks on 3 levels of calculation complexity. N: native wins, C: containered wins, ∼: close to similar results, ?: setup dependent.

	CPU	RAM	I/O
L1 - micro (custom)	N	C	?
L2 - micro (standard)	∼	∼	?
L3 - pipeline	∼		

Our most important findings are summed up on Table 1. On the table 'N' means that *Native run*s resulted slightly better performance according to our indicators, 'C' is for cases when *Containered run*s win, '∼' means no major differences in performance between the two, while '?' means that according to our findings the result of comparison highly depends on the initial setup and the chosen indicator of performance.

References

[1] C. Arango, R. Dernat, J. Sanabria: *An Overview of the Singularity Project*, ArXiv abs/1709.10140 (2017), doi: https://doi.org/10.48550/arXiv.1709.10140.

[2] C. Arango, R. Dernat, J. Sanabria: *Performance Evaluation of Container-based Virtualization for High Performance Computing Environments*, Microsoft Research WA 98052 (2005).

[3] N. G. Bachiega, P. S. L. Souza, S. M. Bruschi, S. d. R. S. de Souza: *Container-Based Performance Evaluation: A Survey and Challenges*, in: 2018 IEEE International Conference on Cloud Engineering (IC2E), 2018, pp. 398–403, doi: 10.1109/IC2E.2018.00075.

[4] A. Torrez, T. Randles, R. Priedhorsky: *HPC Container Runtimes have Minimal or No Performance Impact*, in: 2019 IEEE/ACM International Workshop on Containers and New Orchestration Paradigms for Isolated Environments in HPC (CANOPIE-HPC), 2019, pp. 37–42, doi: 10.1109/CANOPIE-HPC49598.2019.00010.