Result Verification of Engineering Simulations and Transcriptomics Analyses on processor ARM Graviton3

Xavier Besseron and Aurélien Ginolhac

University of Luxembourg

1 Introduction

Scientific software and computers evolve together, software has to be continuously adapted to new platforms to guarantee correct results and achieve the best performance. Software testing is a notoriously arduous task [2] and scientists have to improve their software and port them to new architectures and benefit from better performance. This raises complex issues related to numerical reproducibility [1], which are worsened on HPC platforms and challenges the usual software engineering practices, such as testing and verification.

In the race toward faster and more energy-efficient processors, the AWS Graviton3 based on the ARM Neoverse-V1 architecture is a serious candidate. The performance of this processor has been demonstrated with numerous applications, but, the reproducibility of the results, compared to the 'traditional' architecture x86_64, is rarely mentioned. There is no reason to believe that this processor produces incorrect results. It is well known that numerical reproducibility cannot always be guaranteed even within the same architecture. This brings legitimately the following questions: Will a scientific application, developed on an x86_64-based platform, provide the same results on an ARM platform? Can the differences be quantified and the results verified nevertheless? The answers to these questions are essential to building the trust of the scientists in this new platform.

With this work, we focus on the verification of results produced with the Graviton3 for two HPC applications developed at the University of Luxembourg and initially tested on x86_64 architectures: XDEM, for multi-physics simulation of granular particles; and a Transcriptomics Analysis Workflow for RNA sequencing analysis. These software are frequently used by researchers to produce scientific results and publications. Beyond a simple validation and verification, we want to quantify the numerical differences in comparison with the 'traditional' x84_64 architecture and verify that the scientific conclusions inferred from these results stay the same.

2 Applications and Methodology

The two selected applications hold different characteristics and are complementary for this study. The **eXtended Discrete Element Method (XDEM)** [6] is a simulation framework for the motion and thermal conversion of granular particles, coupled with Computational Fluid Dynamics (CFD). It has many industrial applications including the combustion of biomass, transport phenomena in blast furnace raceways, and selective laser melting for additive manufacture. XDEM highly depends on floating-point operations and numerical errors (or differences) can easily accumulate over the iterations. A **Transcriptomics Analysis workflow**: RNA-seq encompasses numerous steps from sequenced reads to contrasts between condition and quality controls. This workflow[1] relies on Snakemake[5] that connects dependencies and run each step within Singularity containers. Genomic analysis mainly relies on integer arithmetic and logical operations, and its algorithms do not tend to propagate numerical variability over iterations.

We take the point of view of the application developer who has to compile, install and verify the application for the users. The two applications are installed and tested on two platforms: the Aion cluster of the University of Luxembourg[2] based on the processor AMD Epyc ROME 7H12, and the Amazon EC2 instance `c7g.16xlarge`[3] powered by the ARM processor AWS Graviton3. The software is compiled with `-O3` and in complement, we toggle other different compilation flags that can have an impact on the numerical results: `-march=native` to enable all the micro-architecture instructions, and `-ffp-contract=off` to disable floating-point contraction used in the fused multiply-add (FMA) operations. Then, we execute the testsuite and the classic benchmarks for these applications and compare the numerical results produce on the different processors.

3 Particle Trajectories in XDEM

We study the trajectory of particles with a simple test where 27 particles fall on the plate colliding with each other over 1.7s of simulation. Figure 1 shows that the trajectories of some particles quickly diverge between Aion and AWS after a few collisions. We also compare the results for different compilation flags in Figure 2 by plotting the difference of the positions for each particle (as a 'distance'). We only manage to achieve bit-to-bit identical results between x86_64 and ARM when disabling the FMA instructions. However, the two processors using FMA do not generate identical results either.

4 Biomass Drying with coupled XDEM-OpenFOAM

We consider the drying of a biomass as described in [4]. The coupled simulation with XDEM-OpenFOAM is composed of 2667 static particles traversed by a gas flow. We calculate the loss of moisture and compare it with

[1] `https://gitlab.lcsb.uni.lu/aurelien.ginolhac/snakemake-rna-seq`
[2] `https://hpc-docs.uni.lu/systems/aion/`
[3] `https://aws.amazon.com/fr/ec2/instance-types/c7g/`

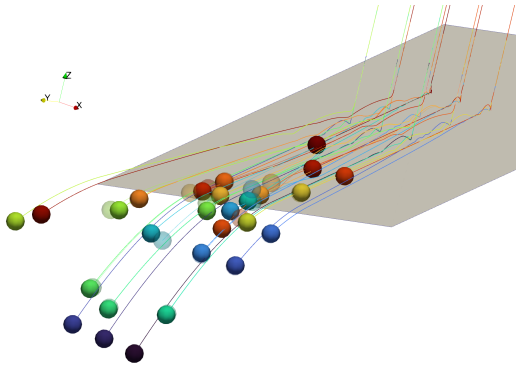

Figure 1: Comparison of particle trajectories between Aion (bright color) and AWS (light color).

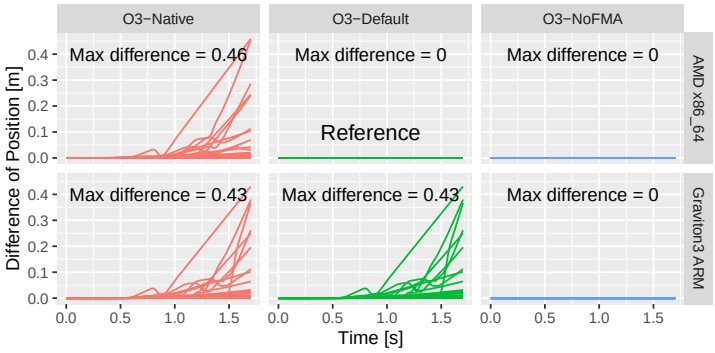

Figure 2: Difference of the positions of the 27 particles for different processor and compilation flags (in comparison with Aion/Default).

experimental observations for validation in Figure 3: the simulations between them cannot be distinguished and have a Mean Absolute Error (MAE) of 2.6e-2 with the experimental data. The Figure 4 shows a difference between the processors with a magnitude of 10e-7, significantly lower than the MAE with the observations. Interestingly, the differences between the simulations occur during the transient period. The results are identical at the beginning and the end of the simulations.

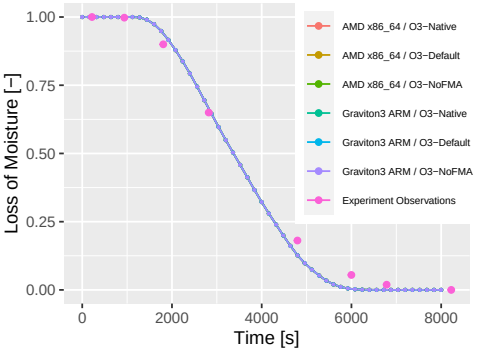

Figure 3: Validation of biomass drying.

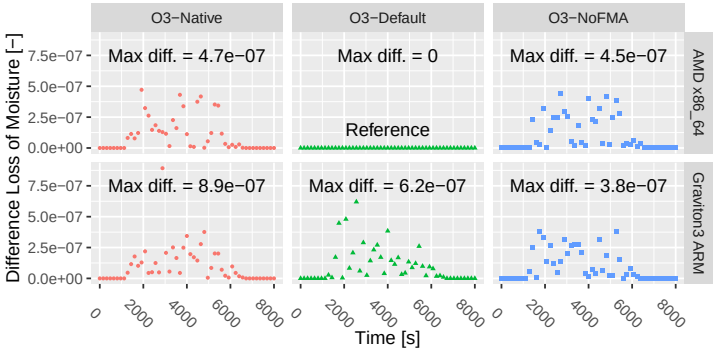

Figure 4: Difference in the simulation results.

5 Transcriptomics Analysis

Regarding the Transcriptomics Analysis, all steps from the raw files to the alignments are strictly identical using the Singularity images on both architectures. Meaning that the mean gene expressions before contrasts are the same. Tiny numerical differences were observed at the differential expression step Figure 5 (only 2 of the 5 metrics are shown but behaved similarly), where the contrast between conditions is performed by empirical Bayesian estimation [3]. However, those tiny differences are not changing the genes ranking, nor the magnitude of fold changes preserving the main scientific conclusions. It would be interesting to the test the compiler flag identified in 3 to see if we retrieved the exact same values.

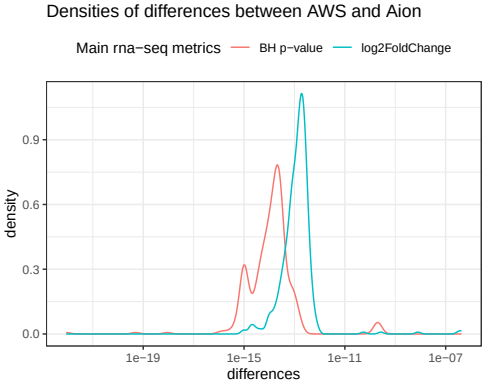

Figure 5: RNA-seq comparison.

References

[1] T. C. Belding. “Numerical Replication of Computer Simulations: Some Pitfalls and How To Avoid Them”. 2000. URL: http://arxiv.org/abs/nlin/0001057.

[2] T. L. Clune and R. B. Rood. “Software Testing and Verification in Climate Model Development”. In: *IEEE Softw.* (2011). DOI: 10.1109/MS.2011.117.

[3] M. I. Love, W. Huber, and S. Anders. “Moderated Estimation of Fold Change and Dispersion for RNA-seq Data with DESeq2”. In: *Genome Biol.* (2014). DOI: 10.1186/s13059-014-0550-8.

[4] A. H. Mahmoudi, F. Hoffmann, and B. Peters. “Application of XDEM as a Novel Approach to Predict Drying of a Packed Bed”. In: *International Journal of Thermal Sciences* (2014). DOI: 10.1016/j.ijthermalsci.2013.07.016.

[5] F. Mölder et al. “Sustainable Data Analysis with Snakemake”. In: *F1000Research* (2021). DOI: 10.12688/f1000research.29032.2.

[6] B. Peters et al. “XDEM Multi-Physics and Multi-Scale Simulation Technology: Review of DEM–CFD Coupling, Methodology and Engineering Applications”. In: *Particuology* (2019). DOI: 10.1016/j.partic.2018.04.005.