The Journal of Open Source Software

TheMatrix: An automated cross-platform benchmarking suite

Rhodri Nelson[*1], **Fabio Luporini**[2], **Mathias Louboutin**[3], **George Bisbas**[4], **and Gerard Gorman**[1]

1 Department of Earth Science and Engineering, Imperial College London, London, SW7 2AZ, UK **2** Devito Codes **3** Georgia Institute of Technology, School of Earth and Atmospheric Sciences, Atlanta, GA **4** Department of Computing, Imperial College London, London, SW7 2AZ, UK

DOI: 10.21105/joss.0XXXX

Software

- Review
- Repository
- Archive

Editor: Editor Name

Submitted: 01 January 1900
Published: 01 January 3030

Summary

TheMatrix is an automated, cross-platform benchmarking suite for the domain-specific language (DSL) and compiler Devito (Louboutin et al., 2019; Luporini et al., 2018). By the push of a button, through utilizing GitHub Actions in tandem with Airspeed Velocity (ASV), Devito's benchmarking tools and computing platforms, a spectrum of benchmarks and regression tests can quickly be executed and compared across a broad range of hardware configurations.

Background

Quantitative performance analysis through benchmarking is an essential method for analyzing the suitability of existing and aiding the design of new, computer systems. Owing to the proliferation of modern architectures and the specializations required to 'correctly' benchmark each specific architecture, cross-platform benchmarking is challenging.

DSL's aimed at high-performance computing offer a compelling route to address this issue. Devito, for example, is a DSL for explicit finite-difference/stencil-based computation on structured grids. At the user level, all Devito operations (finite-difference stencils, boundary conditions, interpolations, cross-correlation, etc.) are composed in `Python`, in a symbolic form based on `SymPy`, and jit-compiled into a dynamic C–library for the specific hardware at hand with the compiler of choice (GCC, Clang, icc, PGI, etc.). Devito natively supports a range of architectures including X64/X86, Power, ARM as well as GPU systems (currently NVidia and AMD). Importantly, the *user code* for any particular simulation is identical for all supported platforms, languages and parallel programming models (e.g. C, MPI, OpenMP, OpenACC).

Exploiting Devito's cross-platform support in tandem with GitHub Actions, TheMatrix has been developed as a tool to quickly, and easily, benchmark the performance on multiple hardware configurations for a range of production-level stencil-based computations. The aims of this software are:

- Assess the performance of stencil-codes across a range of architectures.
- Monitor performance regression on a range of benchmarks.
- Provide a robust and reproducible environment for benchmarking.
- Automate benchmarking across many different application configurations, execution environments and architectures.

*Corressponding author.

The initial set of benchmarks implemented are seismic imaging focused and can be accessed at the following link. As discussed later in the Future Developments section, benchmarks covering a wider variety of physics and related problems (e.g. hydrodynamics) will be added as they are developed.

The authors are currently unaware of a comparable tool for cross-platform benchmarking. In short, TheMatrix:

- Features a fully automated workflow.
- Reuses CI and Cloud infrastructure.
- Is straightforward to extend to benchmarks and architectures.

Despite its tight coupling with Devito, the core ideas behind TheMatrix are applicable to other fields. The aim, in the longer term, is to create a layer of abstraction between the applications/benchmarks to be deployed (today represented by Devito) and the actual TheMatrix, that is the software responsible for the deployment, execution, profiling, and visualization.

Workflow

In this section, the workflow of the TheMatrix and the role of each component is summarized. The key to automating the benchmarking process lies in the interplay between TheMatrix's central configuration file and GitHub Actions. This configuration file, `thematrix.json`, contains the details of each hardware configuration along with which benchmarks each configuration should execute. Once a target architecture is configured[1], a new entry in `thematrix.json` detailing this hardware is all that is required for the benchmarks to be executed on this new hardware.

GitHub Actions

Github Actions is a tool for executing customizable workflows within a repository. *Workflows* contain jobs that are executed in parallel by default which can, in turn, utilize various *actions* to perform specific tasks. In TheMatrix, one such action implements the execution of the whole benchmark suite for a specific combination of:

- Architecture (e.g., Intel Xeon, AMD, GPU, distributed memory).
- Parallel configuration (e.g. MPI, OpenMP, OpenACC).
- Thread/process pinning strategy.
- JIT compiler (e.g. GCC, icc, PGI).

Originally designed to be a high-level interface for continuous integration, GitHub Actions provides a wide range of automation tools for configuring and running repetitive tasks for a tabled set of parameters. Figure 1 outlines TheMatrix's *workflow* configuration.

[1] Configuration steps are outlined in Appendix A.

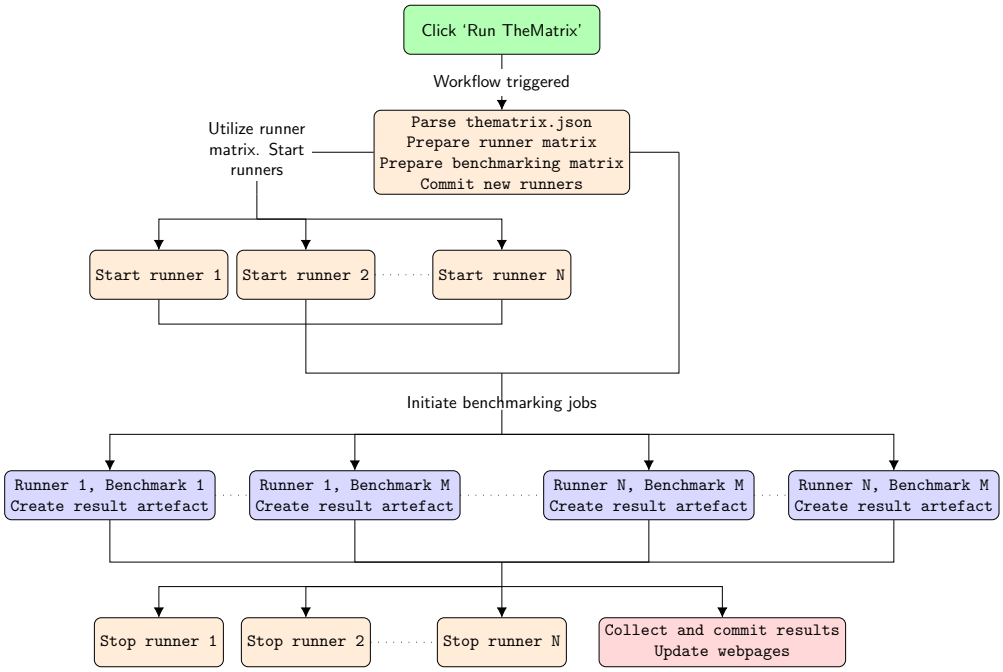

Figure 1: Outline of TheMatrix workflow. The green node indicates an Actions repository dispatch to trigger a benchmarking run, light orange nodes indicate jobs carried out by Github hosted runners, purple nodes indicate benchmarking runs (carried out on the hardware configurations listed on TheMatrix webpage), and the pink node indicates jobs carried out by a self-hosted runner.

Benchmarking and ASV

Devito provides a benchmarking harness for a range of seismic imaging operators (e.g., isotropic/anisotropic acoustic/elastic wave propagators). This framework produces all the fundamental metrics needed for performance evaluation – execution time, GFlops/s, GPoints/s. In TheMatrix we use ASV, a generic tool for benchmarking Python packages, as a wrapper around Devito's benchmarking framework. The role of ASV is to trigger the benchmarks, store the raw results, generate plots, and provide mechanisms for reproducibility (e.g., by tagging each run with information about the software versions used).

An example result for the 3D isotropic acoustic benchmark is shown in Figure 2.

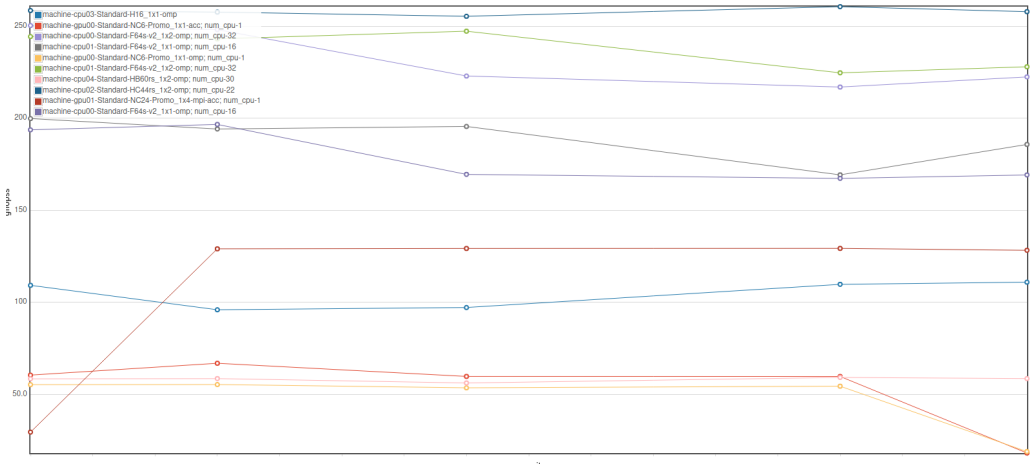

Figure 2: 3D Isotropic acoustic GFlops/s example result. Each line represents a specific architecture, the name of which is given in the legend.

Hardware (The Azure Cloud)

Cloud computing services such as Microsoft's Azure offer convenient access to a range of on-demand hardware configurations. Since TheMatrix workflow provisions and deallocates target hardware on demand, benchmarking a wide range of hardware becomes significantly more affordable (pay-as-you-go Cloud paradigm). Further, owing to the current interest, and the recent increase in companies utilizing cloud-computing for high-performance compute purposes, benchmarks on cloud-based systems are of particular interest. A table of the platforms currently tested can be found here.

TheMatrix is, however, not restricted to cloud platforms in any way; for example, hardware from on-site clusters will soon be incorporated within TheMatrix.

Future developments

TheMatrix currently executes a range of Devito seismic imaging based benchmarks on Azure. The range of architectures benchmarked will be expanded to support benchmarking on other platforms. The addition of like-for-like comparisons across different architectures and platforms provide an additional valuable resource for IT decision-makers.

The range of benchmarks will also be expanded. CloverLeaf style CFD benchmarks are in development, and any parties interested in deploying a particular benchmark are encouraged to contribute.

Finally, as mentioned above, the general framework developed for TheMatrix is not restricted to using Devito: we are currently discussing with other projects, e.g. Firedrake, how this infrastructure can be adapted for their benchmarking and testing purposes. Eventually we aim to develop a layer of abstraction such that this technology can incorporate benchmarks from a range of sources.

Appendix A

From a user perspective, to run the benchmarking suite on a target hardware configuration, the following steps must be executed:

- Configure the target hardware: Install ASV 0.5+, virtualenv, python3+, pip GCC/icc and MPICH (or Open MPI).
- Add the target hardware as a GitHub self-hosted runner – or allow Github Actions to communicate with the hardware through some other means, e.g. an action utilizing `ssh`.
- Add an entry for the runner to the configuration file `thematrix.json`.
- `Run TheMatrix` (GitHub Actions repository dispatch).

The final step of executing the benchmarking workflow is where the automation is embedded. Files relevant to any new configuration or benchmark are created and committed to the GitHub repository. `README .md` and all graphs are updated to include the latest results (and including any new configurations).

Acknowledgements

The authors would like acknowledgement support fom EPSRC EP/R029423/1 and the Open Devito consortium (BP, DUG, Microsoft, Shell).

References

Louboutin, M., Lange, M., Luporini, F., Kukreja, N., Witte, P. A., Herrmann, F. J., Velesko, P., et al. (2019). Devito (v3.1.0): An embedded domain-specific language for finite differences and geophysical exploration. *Geoscientific Model Development*, *12*(3), 1165–1187. doi:10.5194/gmd-12-1165-2019

Luporini, F., Lange, M., Louboutin, M., Kukreja, N., Hückelheim, J., Yount, C., Witte, P., et al. (2018). Architecture and performance of devito, a system for automated stencil computation. *CoRR*, *abs/1807.03032*. Retrieved from http://arxiv.org/abs/1807.03032