GC11-solidearth-15, updated on 23 May 2024
https://doi.org/10.5194/egusphere-gc11-solidearth-15
Galileo Conference: Solid Earth and Geohazards in the Exascale Era

Using Julia for the next generation of HPC-ready software for geodynamic modelling

Albert de Montserrat Navarro[1], Boris Kaus[2], Ludovic Räss[3], Ivan Utkin[3], and Paul Tackley[1]
[1]Institute of Geophysics (ERDW), ETH Zürich, Switzerland
[2]Institute of Geosciences, Johannes-Gutenberg University Mainz, Germany
[3]Laboratory of Hydraulics, Hydrology and Glaciology (VAW), ETH Zürich, Switzerland

Following the long-standing paradigm in HPC, scientific software has been typically written in high-level statically typed and compiled languages, namely C/C++ and Fortran. The arguably low productivity rates of these languages led to the so-called *two-language problem*, where dynamic languages such as Python or MATLAB are used for prototyping purposes, before porting the algorithms to high-performance languages. The Julia programming language aims at bridging the productivity rates and other advantages of such dynamic languages without sacrificing the performance provided by their static counterparts. The combination of this high performance, productivity rates and other powerful tools, such as advanced meta-programming (i.e. code generation), make Julia a suitable candidate for the next generation of HPC-ready scientific software.

We introduce the open-source and Julia-written package JustRelax.jl (https://github.com/PTsolvers/JustRelax.jl) as a way forward for the next generation of geodynamic codes. JustRelax.jl is a production-ready API for a collection of highly-efficient numerical solvers (Stokes, diffusion, etc.) based on the embarrassingly parallel pseudo-transient method. We rely on ParallelStencil.jl (https://github.com/omlins/ParallelStencil.jl), which leverages the advanced meta-programming capabilities of Julia to generate efficient computational kernels agnostic to the back-end system (i.e. Central Processing Unit (CPU) or Graphics Processing Unit (GPU)). Using ImplicitGlobalGrid.jl (https://github.com/eth-cscs/ImplicitGlobalGrid.jl) to handle the MPI and CUDA-aware MPI communication, these computational kernels run seamlessly in local shared-memory workstations and distributed memory and multi-GPU HPC systems with little effort for the front-end user.

Efficient computation of the (local) physical properties of different materials is another critical feature required in geodynamic codes, for which we employ GeoParams.jl (https://github.com/JuliaGeodynamics/GeoParams.jl). This package provides lightweight, optimised, and reproducible computation of different material properties (e.g. advanced rheological laws, density, seismic velocity, etc.), amongst other available features. GeoParams.jl is also carefully designed to support CPU and GPU devices, and be fully compatible with other external packages, such as ParallelStencil.jl and existing auto-differentiation packages.

We finally show high-resolution GPU examples of geodynamic models based on the presented open-source Julia tools.