Hessisches Kompetenzzentrum
für Hochleistungsrechnen

Performance Analysis of the Ice-Sheet and Sea-Level System Model (ISSM)

Project Manager
Yannic Fischler

Researchers
Peter Arzt, Sven Donnerhak, Jonas Kubicki, Manuel Senge, Georg Kurt Bettenhausen, Marcel Mittenbühler and Martin Rückamp

Principal Investigator
Prof. Dr. Christian Bischof

Project Term
2021 - 2022

Clusters
Lichtenberg II Cluster Darmstadt

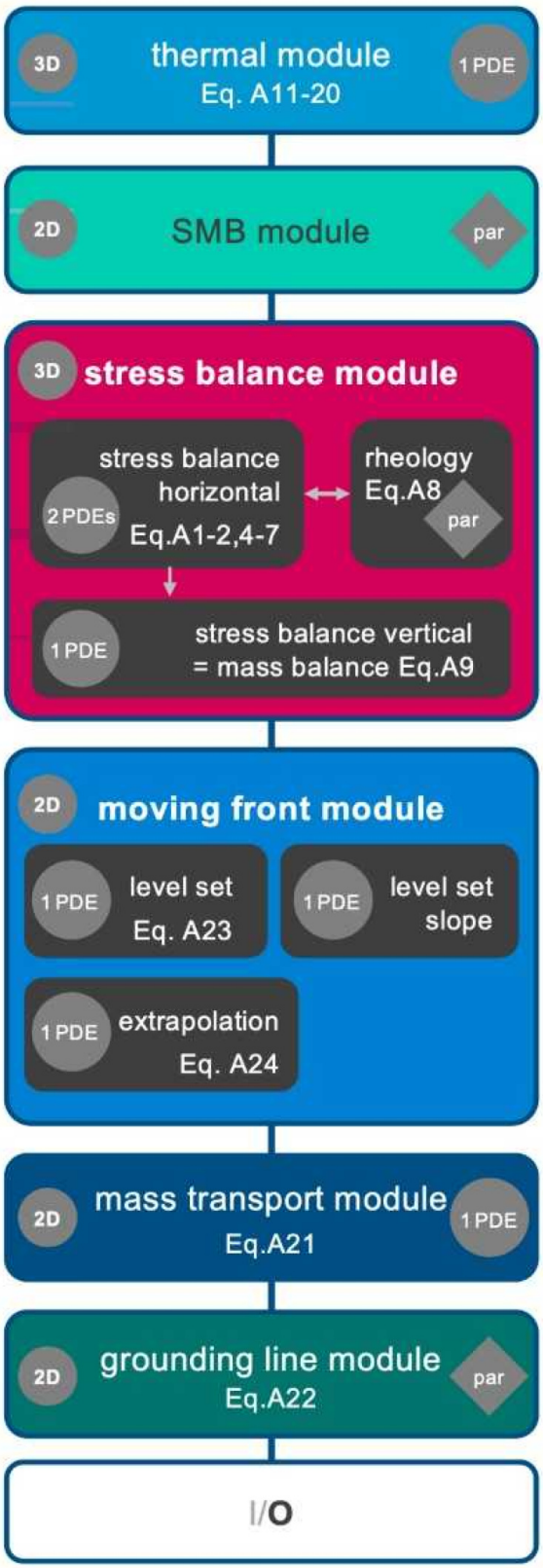
Software
Python

Additional Software
ISSM, GCC, OpenMPI, PETSc, Score-P, cube, clang, PIRA, OpenTuner

Institute
Department of Computer Science

University
Technische Universität Darmstadt

JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN
TECHNISCHE UNIVERSITÄT DARMSTADT
UNIKASSEL VERSITÄT
Philipps Universität Marburg



JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN
TECHNISCHE UNIVERSITÄT DARMSTADT
UNIKASSEL VERSITÄT
Philipps Universität Marburg

Introduction

Earth modeling is a wide field of different scientific domains, e.g., ice-modeling. All domains have in common, that they need to be able to handle large areas. Therefore, scalability is a significant property of earth models as they compute large domains and long time periods. The Ice-sheet and Sea-level System Model (ISSM) is a well-known framework, which is used to simulate large ice-sheets as the Greenland ice-sheet. It is written in C++ and employs process parallelism (MPI) through PETSc. Ice is a very complex material with different physical properties. It behaves like a fluids and solids at the same time. Additionally, it is dependent from different external forces like weather, temperature and the bedrock topology. The complex multi physics model of ISSM is implemented in individual modules running in a predefined sequence (see Figure 1).
In this project, we analyzed the scaling of ISSM running a realistic Greenland setup on the Lichtenberg HPC System. We aimed to identify the sweet spot (number of processes with which the code runs fastest). We also put a focus on the load balancing of the code and identified computational hotspots and optimization potential.

Methods

We instrumented the code using Score-P with a manually created filter. We run the instrumented code with a Greenland setup with minimal edge length of 250 m (G250). Thereby we got low overhead runtime measurements of ISSM on 96 up to 6144 MPI processes on up to 128 nodes.
Furthermore, we developed PIRA-LID, a tool based on PIRA and clang, for automatic instrumentation of load imbalanced regions in MPI applications and applied it on ISSM. Finally, we identified load imbalanced regions and tried to identify optimal configuration by automatic optimization using OpenTuner.

Results

We show, that large setups as the G250 provide enough computational load to take advantage of up to 3000 MPI processes within the time stepping sequence of ISSM (Figure 3). The overall scaling is limited by the moving front module and within the modules the assembly of the equation systems is the major issue. Therefore we present the timing data of the matrix assembly per module in Figure 2. In general, the most costly module is the stress balance, even while we run it using the higher order approximation, which is known to be much faster than full stokes.
The load imbalance detection shows, that the creation of the equation system is load imbalanced, because some processes process grid elements, which do not include ice (nothing to do) and other has to process the ice regions. Our optimizer was able to improve these regions, but the whole runtime did not improve, because other code regions became load imbalanced by our changes. Ultimately, adjusted load balancing parameters

JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN
TECHNISCHE UNIVERSITÄT DARMSTADT
UNIKASSEL VERSITÄT
Philipps Universität Marburg

did not show any significant impact on the overall runtime.

Discussion

The scalability of ISSM using up to 3000 MPI processes on realistic setups is sufficient to run it on HPC systems as the Lichtenberg 2. The load balancing is not optimal, but already sufficient. The individual modules of the code have different dependencies and an optimal load balancing in all modules simultaneously seams not to be possible.
To improve the throughput of ISSM we suggest to focus on the scaling bottlenecks of the moving front module and the node level performance of the creation of the equation system within the stress balance module.

Figures

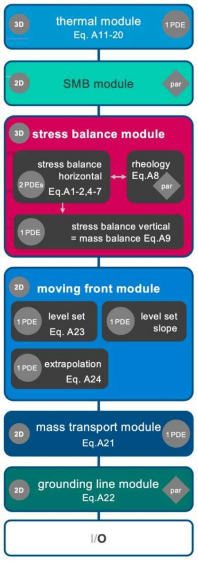


Figure 1: The sequence of modules in a transient time step in ISSM. Small grey circles indicate the dimension of the equations of the module (3D, 2D). Larger grey circles with PDE are denoting if and how many partial differential equations are solved. Diamonds with 'par' indicate that only an algebraic equation is evaluated. On the right side, we list the DOF for particular mesh resolutions and modules.

JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN
TECHNISCHE UNIVERSITÄT DARMSTADT
UNIKASSEL VERSITÄT
Philipps Universität Marburg

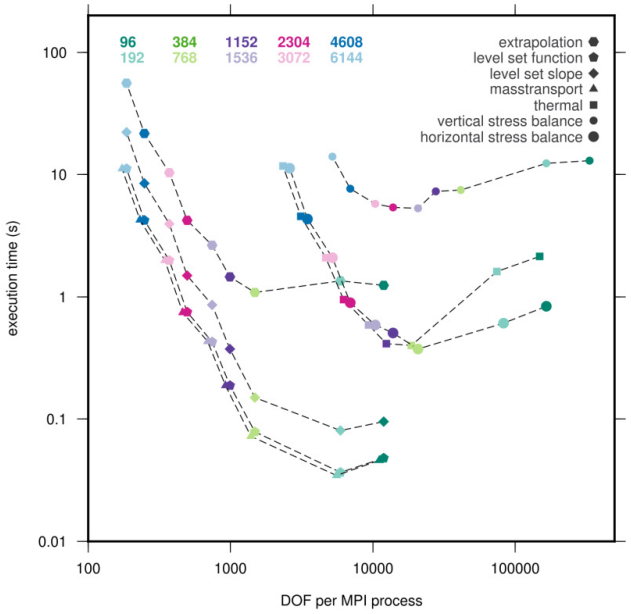


Figure 2: Execution time for matrix assembly for various modules. Symbols are representing the modules, while the color is denoting the number of MPI processes.

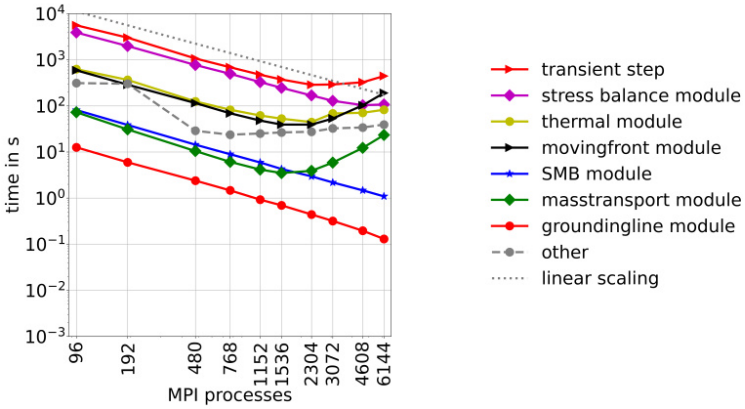


Figure 3: Run-time of a transient time step of ISSM without output handling in resolution G250.

Publications

Fischler, Y; Rückamp, M; Bischof, C.; Aizinger, V; Morlighem, M and Humbert, A. A scalability study of the Ice-sheet and Sea-level System Model (ISSM, version 4.18), Geoscientific Model Development (2022) https://doi.org/10.5194/gmd-15-3753-2022

Arzt, P; Fischler, Y; Lehr, J-P; Bischof, C. Automatic Low-Overhead Load-Imbalance Detection in MPI Applications, Lisbon, Portugal (2021) https://doi.org/10.1007/978-3-030-85665-6_2

Last Update: 2023-08-26 17:12