EGU2020-18749
https://doi.org/10.5194/egusphere-egu2020-18749
EGU General Assembly 2020

Towards a scalable framework for earth science simulation models, using asynchronous many-tasks

Kor de Jong[1,2], Derek Karssenberg[1], Deb Panja[2], and Marc van Kreveld[2]
[1]Utrecht University, Faculty of Geosciences, Department of Physical Geography, Utrecht, The Netherlands
(k.dejong1@uu.nl)
[2]Utrecht University, Faculty of Sciences, Department of Information and Computing Sciences, Utrecht, The Netherlands

Computer models are built with a specific purpose (or scope) and runtime platform in mind. The purpose of an earth science simulation model might be to be able to predict the spatio-temporal distribution of fresh water resources at a continental scale, and the runtime platform might be a single CPU core in a single desktop computer running one of the popular operating systems. Model size and complexity tend to increase over time, for example due to the availability of more detailed input data. At some point, such models need to be ported to more powerful runtime platforms, containing more cores or nodes that can be used in parallel. This complicates the model code and requires additional skills of the model developer.

Designing models requires the knowledge of domain experts, while developing models requires software engineering skills. By providing facilities for representing state variables and a set of generic modelling algorithms, a modelling framework makes it possible for domain experts without a background in software engineering to create and maintain models. An example of such a modelling framework is PCRaster [3], and examples of models created with it are the PCRGLOB-WB global hydrological and water resources model [2], and the PLUC high resolution continental scale land use change model [4].

Models built using a modelling framework are portable to all runtime platforms on which the framework is available. Ideally, this includes all popular runtime platforms, ranging from shared memory laptops and desktop computers to clusters of distributed memory nodes. In this work we look at an approach for designing a mod elling framework for the development of earth science models using asynchronous many-tasks (AMT). AMT is a programming model that can be used to write software in terms of relatively small tasks, with dependencies between them. During the execution of the tasks, new tasks can be added to the set. An advantage of this approach is that it allows for a clear separation of concerns between the model code and the code for executing work. This allows models to be expressed using traditional procedural code, while the work is performed asynchronously, possibly in parallel and distributed.

We designed and implemented a distributed array data structure and an initial set of modelling algorithms, on top of an implementation of the AMT programming model, called HPX [1]. HPX provides a single C++ API for defining asynchronous tasks and their dependencies, that execute

locally or on remote nodes. We performed experiments to gain insights in the scalability of the individual algorithms and simple models in which these algorithms are combined.

In the presentation we will explain key aspects of the AMT programming model, as implemented in HPX, how we used the programming model in our framework, and the results of our scalability experiments of models built with the framework.

References

[1] HPX V1.3.0. http://doi.acm.org/10.1145/2676870.2676883, 5 2019.

[2] E. H. Sutanudjaja et al. PCR-GLOBWB 2: a 5 arc-minute global hydrological and water resources model. Geoscientific Model Development Discussions, pages 1–41, dec 2017.

[3] The PCRaster environmental modelling framework. https://www.pcraster.eu

[4] PLUC model. https://github.com/JudithVerstegen/PLUC_Mozambique