Geophysical Research Abstracts
Vol. 19, EGU2017-14973, 2017
EGU General Assembly 2017

Scalable geocomputation: evolving an environmental model building platform from single-core to supercomputers

Oliver Schmitz, Kor de Jong, and Derek Karssenberg
Utrecht University, Faculty of Geosciences, Physical Geography, Utrecht, The Netherlands (o.schmitz@uu.nl)

There is an increasing demand to run environmental models on a big scale: simulations over large areas at high resolution. The heterogeneity of available computing hardware such as multi-core CPUs, GPUs or supercomputer potentially provides significant computing power to fulfil this demand. However, this requires detailed knowledge of the underlying hardware, parallel algorithm design and the implementation thereof in an efficient system programming language. Domain scientists such as hydrologists or ecologists often lack this specific software engineering knowledge, their emphasis is (and should be) on exploratory building and analysis of simulation models. As a result, models constructed by domain specialists mostly do not take full advantage of the available hardware. A promising solution is to separate the model building activity from software engineering by offering domain specialists a model building framework with pre-programmed building blocks that they combine to construct a model. The model building framework, consequently, needs to have built-in capabilities to make full usage of the available hardware.

Developing such a framework providing understandable code for domain scientists and being runtime efficient at the same time poses several challenges on developers of such a framework. For example, optimisations can be performed on individual operations or the whole model, or tasks need to be generated for a well-balanced execution without explicitly knowing the complexity of the domain problem provided by the modeller. Ideally, a modelling framework supports the optimal use of available hardware whichsoever combination of model building blocks scientists use. We demonstrate our ongoing work on developing parallel algorithms for spatio-temporal modelling and demonstrate 1) PCRaster, an environmental software framework (http://www.pcraster.eu) providing spatio-temporal model building blocks and 2) parallelisation of about 50 of these building blocks using the new Fern library (https://github.com/geoneric/fern/), an independent generic raster processing library. Fern is a highly generic software library and its algorithms can be configured according to the configuration of a modelling framework. With manageable programming effort (e.g. matching data types between programming and domain language) we created a binding between Fern and PCRaster. The resulting PCRaster Python multicore module can be used to execute existing PCRaster models without having to make any changes to the model code. We show initial results on synthetic and geoscientific models indicating significant runtime improvements provided by parallel local and focal operations. We further outline challenges in improving remaining algorithms such as flow operations over digital elevation maps and further potential improvements like enhancing disk I/O.