Geophysical Research Abstracts
Vol. 21, EGU2019-13693, 2019
EGU General Assembly 2019

Analysis of domain decomposition methods for distributed land surface models on HPC-Systems

Alexander von Ramm, Wolfgang Kurtz, and Jens Weismüller
Leibniz Rechenzentrum, Environmental Computing, Garching, Germany (alexander.vonramm@lrz.de)

Different hydrological research projects currently tackle questions which require high-resolution land surface models on a global scale. The computational demands of high-resolution models on this scale is beyond the capabilities of modern desktop workstation or small distributed clusters. Deploying these models on HPC-Systems requires the efficient distribution of computational load among available computing resources and a communication strategy. The distribution of computational load is not a simple task, because it stems mainly from the evaluation of highly parameterized equations. Therefore it varies in both space and time. For an efficient and well scalable code the load-balancing strategy and the communication strategy need to considered in a joint manner.
In this contribution, we investigate four different domain decomposition algorithms with respect to their suitability for distributed land surface models. We focus on the algorithms ability to evenly distribute computational load among available compute cores and how well the partitioning found integrate with the communication scheme dictated by land surface models. We consider three coordinate-based algorithms as well as one graph-based algorithm. The coordinate-based algorithms include block-wise domain decomposition, recursive coordinate bisection as well as domain decomposition based on the Hilbert space filling curve. The graph-based algorithm is employed to partition the flow-direction graph. We perform artificial strong scaling experiments from 1 to 8192 processors for a PROMET model of the Upper-Danube.
All methods considered can distribute the computational load among available resources with similar quality. The graph-based method excels in reducing the communication overhead. In terms of communication overhead the graph-based method outperforms all other methods by at least a factor of 2 on 8192 processors. On fewer processors the advantage of the graph-based method is even greater.