Managing Iterations Graphs in Multiscale Applications using Akka

Katarzyna Rycerz (1,2), Marian Bubak (1,2)

(1) AGH University of Science and Technology, Institute of Computer Science AGH, Department of Computer Science, al. Mickiewicza 30, 30-059 Kraków, Poland
(2) ACC Cyfronet AGH, Nawojki 11, 30-950, Kraków, Poland

Motivation

- Different coupling templates of multiscale applications [1] supported by building and execution tools [2,3].
- Efficient execution support for iteration graphs of multiscale applications on a fine-grained (iteration level).

Solution

- to group modules of similar demands for the same resources
- the modules do not necessary have to belong to the same application
- the behavior of each module is modeled by Akka actors used to send commands to the wrapping service on the actual resource.
- once send to the same resources, the modules need to be properly queued to execute part of one application in the idle time of the other

Results

- a study with an artificial macro-micro model using Akka actors
- LU factorization algorithm with parametrized matrix size steering the CPU time used by the simulation
- comparison of execution of two macro modules on the same CPU resource with different Akka executor settings
- the results confirmed that appropriate Akka executor mechanisms allows one application to use gaps during the idle time of the other application.

Conclusions

- The preliminary results are promising for building the Akka-based framework managing execution of the applications consisting of distributed modules communicating in iterations.
- As a resources, apart from HPC we also plan to use cloud resources and mechanism of lightweight virtualization containers.
- The main challenges are to build a solution for a set of legacy multiscale applications[4].