Dry Schemas for parallel (RePast-based) ABM simulations

Laszlo Gulyas, Gabor Szemes, Walter de Back, and George Kampis

lgulyas@colbud.hu, ugyanaz@gmail.com, walter@deback.net, gkampis@colbud.hu

Collegium Budapest (Institute for Advanced Study), Hungary

In the modeling of complex systems, the need for validation, as a measure of how well a model approaches reality, naturally arises. One of the important parameters is the number of agents an agent based model (ABM) can handle, i.e., the issue of the scalability of the model. To achieve acceptable scalability, the number of individuals should reach numbers (ideally, millions or more) that cannot be feasibly handled in any stand-alone system. The development of parallel code for cluster-level or inter-cluster computation is inevitable. However, the parallelization of an ABM is typically a hard task, due to the high level of inter-process-communication.

Different approaches can be found in the literature. Without completeness: they range from solutions that use Graphical Processing Units [Lysenko et al 2008], through the automatic parallelization of ABMs [Scheutz 2005] to the techniques applicable to spatial (e.g. ecological) ABMs [Wang et al 2006], [Glass et al 1997] and [Mellott et al 1999]. Many solutions exist, but all these solutions have been implemented in a hard-to-reuse way.

As a consequence of this situation, writing a parallel version of a new ABM simulation is at least as hard as developing the original model, and usually takes an expert of parallel computing. Our aim was to develop a user friendly toolkit to ease the process of ABM parallelization. We provide reusable, parallel simulation schemas (Dry Schemas), which enable modelers to treat their parallel system in almost the same was as they do the original. The paper/talk details our ongoing developments based on the idea that ABMs can be classified on the basis of their interior communication topology, and that this classification can be exploited by developing ready-made parallelization templates. Following up on this categorization, we have developed six reusable parallel simulation schemas in the Java language (using the ProActive toolkit for parallel computing) that can be instantiated with simulation-specific code. Our aim was to give general, domain-independent support for ABM (RePast) modelers, where the parallel piece of code is completely transparent.

The paper/talk details our approach as well as the implementation, and shows performance results, code metrics, and also hints on how some of the templates work in a particular grid environment, the QosCosGrid (QCG, www.qoscosgrid.eu, supported by an EU FP6 grant), the umbrella project for the reported work.