

Extending the COMET component model to support hierarchical composite data : Aevol case study

Jerry Lacmou Zeutouo^(a), Christian Perez^(a), Thierry Gautier^(a), Jonathan Rouzaud Cornabas^(b)

(a) The joint project-team Avalon INRIA, LIP, ENS-Lyon, Lyon, France

jerry.lacmou-zeutouo@inria.fr, christian.perez@inria.fr, thierry.gautier@inrialpes.fr

(b) Univ. Lyon, INSA Lyon, Inria, CNRS, LIRIS, Lyon, France

jonathan.rouzaud-cornabas@inria.fr

Résumé

Numerical simulation is a key technology for many application domains. It is nowadays considered as the third pillar of sciences (with experiment and theory) and is critical to gain competitive position. Thanks to the democratization of high-performance computers (HPC), complex physics, molecular biology, and more generally complex systems can now be routinely simulated. Aevol (<http://aevol.fr/>) is an example of such a simulator. It consists in simulating millions of generations of an evolving population of micro-organisms. Each generation is made up of a set of populations. For each individual, the model simulates how it evolves through stochastic selections and mutations, which consist in randomly modifying its structures. A simulation is characterized by a Petri dish of bacteria (modeled as a two-dimensional array), where each bacterium has cyclic DNA (modeled as an array of characters) with a few thousand to millions of base pairs.

One problem with Aevol, as with many HPC applications, is that it mixes functional parts of interest to bioinformaticians with HPC concerns. As a result, it is very difficult for bioinformaticians to adapt the code to new use cases without the help of HPC experts. COMET, developed by the AVALON Inria team, is a component-based programming model for HPC applications that aims to address this problem. From a parallel dataflow description of the application, it generates the OpenMP code that performs the tasks at runtime, reducing the need for user expertise in HPC. However, the current version of the COMET model is not expressive enough to fully support Aevol. This talk will present current efforts to define and support hierarchical composite data types that will enable the composition of parallel Aevol code that manipulates Petri dishes and DNA. We will present Aevol and its requirements, the envisioned evolutions for COMET with respect to hierarchical composite data types, as well as preliminary results.

Mots-clés : Hierarchical partitioning, Task-based model, Component-based programming model, *in-silico* simulation
