Project Team INRIA: HiePACS

Author of the proposal research subject: *A. Esnard*

Title of the proposal research subject: Dynamic Load Balancing for Massively Parallel Coupled Codes.

Scientific context (10-15 lines):

In the field of scientific computing, the load balancing is an important step conditioning the performance of parallel programs. The goal is to distribute the computational load across multiple processors in order to minimize the execution time. This is especially critical when the number of processors increases, i.e., for massively parallel applications running on many-core architectures. For some scientific applications, whose load evolution is unpredictable (e.g. adaptive mesh refinement), it is required to periodically compute a new balancing at runtime, using a dynamic load balancing algorithm. This is a well-known problem [1,2], that is unfortunately NP-hard. The most common approach to solve it is based on graph or hypergraph partitioning method, using mature and efficient software tools such as Metis [3], Zoltan [4] or Scotch [5].

Nowadays, numerical simulation are becoming more and more complex, mixing several models and codes to represent different physics or scales. Here, the key idea is to reuse available legacy codes through a coupling framework instead of merging them into a standalone application. For instance, the simulation of the earth's climate system typically involves at least 4 codes for atmosphere, ocean, land surface and sea-ice [6]. Combining such different codes are still a challenge to reach high performance and scalability.

Goal (10-15 lines) & Project (10-15 lines)

In this context, one crucial issue is undoubtedly the load balancing of the whole coupled simulation that remains an open question. The goal here is to find the best data distribution for the whole coupled codes and not only for each standalone code, as it is usually done. Indeed, the naive balancing of each code on its own can lead to an important imbalance and to a communication bottleneck during the coupling phase, that can dramatically decrease the overall performance. Therefore, one argues that it is required to model the coupling itself in order to ensure a good scalability, especially when running on tens of thousands of processors. In other words, one must develop new algorithms and software implementation to perform a “coupling-aware” partitioning of the whole application.

Another related problem we plan to investigate is the problem of resource allocation. This is particularly important for the global coupling efficiency, because each code involved in the coupling can be more or less computationally intensive. Consequently, there is an effective trade-off to find between resources assigned to each code in order to avoid that one of them wait for the others at each coupling stage. For instance, if we consider a given number of processors and two coupled codes, the question is how to split the processors among each code. Another question is what happens if one code becomes more computationally intensive? In such a case, it could be convenient to dynamically adapt the number of resources used at runtime. However, most of existing works on repartitioning only consider a fixed number of processors. This can be very inefficient, especially in terms of resource consumption [7].

We expect this work to be really suitable for the next generation of large-scale scientific simulations running on many-core architectures.

Advisors: *A. Esnard, J. Roman*

References (max 5 lines):

1. B. Hendrickson and K. Devine, "Dynamic load balancing in computational mechanics," in Computer Methods in Applied Mechanics and Engineering, vol. 184, 2000, pp. 485–500.
2. J. D. Teresco, K. D. Devine, and J. E. Flaherty, "Partitioning and dynamic load balancing for the numerical solution of partial differential equations," in Numerical Solution of Partial Differential Equations on Parallel Computers, ser. Lecture Notes in Computational Science and Engineering, Eds. Springer, Berlin Heidelberg, 2006, vol. 51, pp. 55–88.
3. Metis: http://www.cs.umn.edu/~metishttp://www.cs.umn.edu/~metis
4. Zoltan: http://www.cs.sandia.gov/Zoltanhttp://www.cs.sandia.gov/Zoltan
5. Scotch: http://scotch.gforge.inria.fr/http://scotch.gforge.inria.fr
6. CESM (Community Earth System Model): http://www.cesm.ucar.edu/http://www.cesm.ucar.edu
7. S. Iqbal, G. Carey, M. Padron, J. Suarez, and A. Plaza, "Load balancing with variable number of processors on commodity clusters," in High Performance Computing Symposium, San Diego, 2002, pp. 135–140.

Keywords (max 5-6 lines): *high-performance parallel computing, numerical simulation, code coupling, dynamic load-balancing, graph partitioning, ...*

Duration: *36 months*