OOPar: An Object Oriented Environment for Implementing Parallel Algorithms

Philippe Devloo
Faculdade de Engenharia Civil, Arquitetura e Urbanismo – UNICAMP, Campinas, SP, Brazil

Abstract

OOPar is a C++ library which implements an interface to MPI (or other communication library) to offer the user a high level interface for implementing parallel algorithms. Large scale software projects have been parallelized using OOPar and their results will be presented during this presentation.

OOPar introduces two concepts to help the programmer in defining a parallel algorithm: distributed data and tasks which act on the distributed data.

Distributed data are objects of classes which can be transmitted between processors. The user of the OOPar library can define his own types by deriving class from the TPZSaveable class and implementing the virtual Read/Write methods. Distributed data objects are managed by an object of type OOPDataManager. Each processor has its OOPDataManager object which administers TPZSaveable objects and the access requests issued by the OOPTask objects.

Task objects implement part of a parallel algorithm acting upon and transforming distributed data objects. Task objects are submitted to a TaskManager object. Each processor has a unique TaskManager object. In order to sequence tasks in the proper order, a version is associated with distributed data objects and tasks depend on a distributed data object with a certain version. After transforming a distributed data object, the task object increments the version of the data object.

Task objects are assigned a processor number. When submitted to the TaskManager on the current processor, the manager will send the task object to the assigned processor using the the serialization mechanism defined by the TPZSaveable class. The parallel execution of the algorithm depends on the distribution of the tasks over the pool of processors.

OOPar is useful for both parallelizing existing software and implementing new algorithms. The environment has been successfully applied to the parallelization of a three dimensional computational fluid dynamics software including turbulence models and explicit/implicit solvers. A parallel implementation of an error estimator has also been implemented using OOPar.

Currently substructuring techniques are being implemented within OOPar with the intent of developing a parallel finite element solver for three dimensional problems.

OOPar is a public domain software project and can be downloaded from our cvs server at our laboratory. Instructions can be found at http://labmec.fec.unicamp.br