Workpackage 2: Conceptual Model

Objectives

The objective of this workpackage is to develop a conceptual model of SSL (the Standard Skeleton Library). This task is particularly important because it describes the philosophy, the design principles, and the envisaged use of the library, thus providing an organizational framework for the rest of the workpackages in the project.

The main challenge of the conceptual model is to find an acceptable trade-off between several contradictory requirements:

- Achieving wide acceptance of SSL among parallel-programming practitioners, most of whom are used to the imperative or object-oriented programming paradigm
- Providing advanced features from functional programming, like higher-order functions, partial application, polymorphism, semantics-preserving transformations, etc.
- Facilitating the automatic portability of SSL-codes across major computing platforms, providing acceptable performance and at the same time resource-tuning options for important classes of architectures
- Identifying and standardizing a number of the most important skeletons, while enabling users to define new application-oriented skeletons

Description of Work

The activities required by this workpackage will include:

- Developing the overall structure of the programming environment around SSL, i.e. the relations and communication between SSL itself, specific APIs, tools, cost models and target machines
- Providing the main envisaged scenarios of how the work of users should be organized and explaining how these maximize ease of programming while at the same time offering portability and high performance
- Identifying and formally specifying the SSL collection of different kinds of skeletons:
  - Data-parallel skeletons
  - Task-parallel skeletons
  - Construction skeletons/rules
- Formulating the main features provided by SSL and restrictions imposed by the library wrt. the following points:
  - Platform independence and performance tuning
  - Programming style (distribution of computation and data, communication, type system, etc.)
  - Performance prediction and cost models
  - Tool support

Deliverables

The principal document developed in this work package , called *SSL-Credo*, will describe:

- the conceptual background and principles of SSL, including the major points mentioned in the above Description of Work
- the main components of the library and their interplay in the process of program development
- SSL's place in the parallel-computing setting and in particular its advantages over existing paradigms like virtual shared-memory programming and pure message-passing programming
- the short- and long-term goals of the SSL project, including possible future extensions for the emerging paradigm of grid computing

The level of detail in the *Credo* document should suffice to demonstrate convincingly how the challenges mentioned in the "Objectives" are met. Further details will be provided in the documents delivered by the other Work Packages.

Milestones and Expected Results

The mentioned document, *SSL Credo*, will be developed iteratively during the planned duration of the project. The main iterations can be anchored to the phases of specification, design, implementation, and evaluation of SSL and will be produced every 6 months:

- *Credo-Spec*: After the first 6 months, the preliminary version of the document will be developed based on the state of the art in the area (latest research papers) and the input of the project participants. This document will contain the list of skeletons included in SSL.
- *Credo-Design*: After 12 months, the document will be extended to include the details of the SSL library under design. In particular, the main features and restrictions as well as envisaged scenarios of use will be described.
- *Credo-Impl*: After 18 months, the document will be enhanced with the details of the implementation and first performance measurements. It will also contain the description of the SSL toolkit.
- *Credo-Final*: Will be delivered by the end of the project. This final document will present the implemented SSL, the lessons learned from the project, open questions and problems and future work.