University of Stuttgart

Institute for Parallel and Distributed High Performance Systems (IPVR)

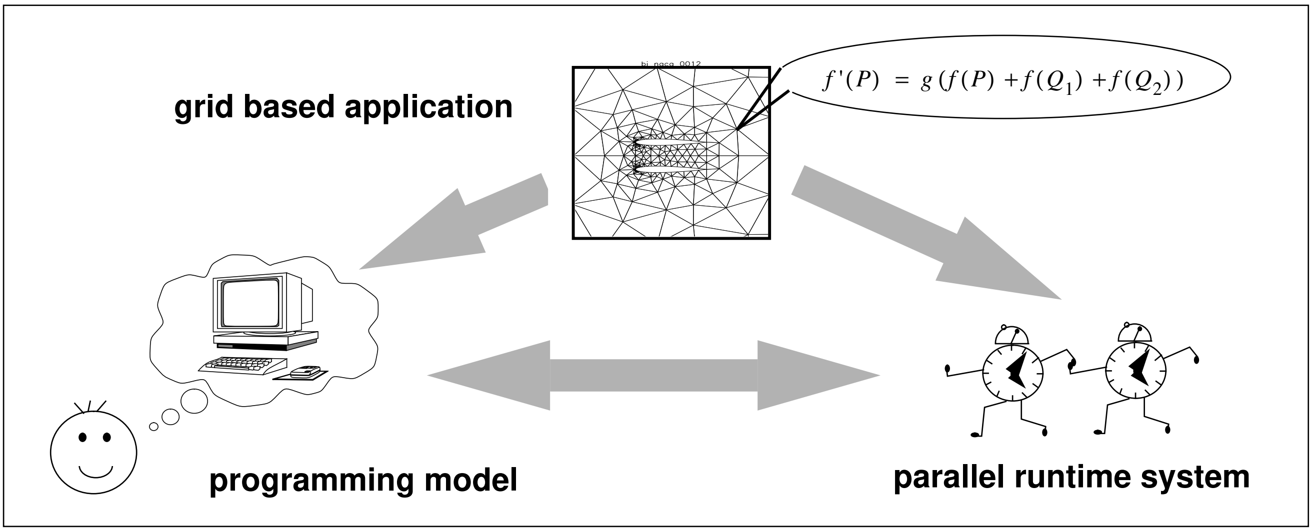
Breitwiesenstr. 20-22, W-7000 Stuttgart 80
Phone: +49 711 7816-448 Fax: +49 711 7816-424

Prof. Dr.-Ing. A. Reuter

GRIDS – A Programming Environment for Grid Based Technical and Scientific Applications on Parallel Systems

GRIDS is a programming environment for technical and scientific applications on parallel systems and workstation networks. It has been developed at the University of Stuttgart in cooperation with IBM Deutschland Informationssysteme GmbH. GRIDS overcomes the difficulties and obstacles of programming grid based methods on parallel computers. In particular irregular grids are supported. GRIDS offers a problem-oriented, architecture independent programming interface. The solution methods are expressed in a common, sequential programming language. The resulting code is parallelized automatically by using knowledge about the structure of grid methods. It is executed highly efficient by the parallel runtime system. GRIDS offers the following advantages for the development of software:

- The compute power offered by parallel computers is usable without explicit parallel programming.
- Administration of parallel execution is carried out by the GRIDS runtime system dynamically adapting to the problem inherent parallelism.
- Applications written with GRIDS are portable over different architectures and generations of parallel computers. GRIDS can exploit special hardware features transparent to the programmer.
- The effort of programming parallel machines is substantially reduced. GRIDS offers a modular construction system for grid based computation.
- Programming is simplified to describing “what to do” instead of coding “how to do it in parallel”.



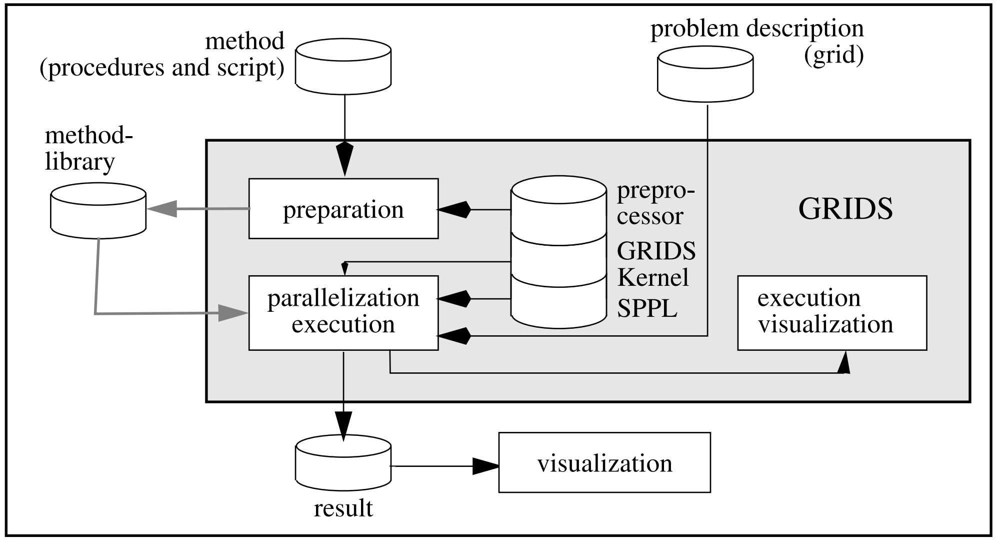
GRIDS opens a way to solve technical problems in a new order of magnitude cost-efficiently by parallel computation.The programming environment facilitates the introduction of parallel computing to this large application area and thereby contributes to the break-through of this innovative technology.

Application area

We present a prototype of GRIDS, a programming system for grid based technical and scientific applications on parallel systems and workstation networks. This important class of applications covers, among others, simulation problems in computational fluid dynamics (CFD), semiconductor technology and circuit design. Problems of this class are suitable for parallelization due to the locality properties of the methods.

Programming interface

The basic idea of the GRIDS programming model is the separation of problem geometry description from algorithm description. The use of a common, sequential programming language with additional, grid method specific constructs simplifies the formulation of the solution algorithm. Coded solution methods can be modularly reused with different problems or problem geometries.



The solution method is described at two levels. At the “local” level the procedures for the iteration steps are programmed. At the “global” level a script is used. The script defines the order of the iteration steps and the parts of the grid they are executed on. The topology description defines the computational grid, consisting of elements of different types (vertices, edges, volumes, ..) and the neighbor relations between these elements.

Runtime system

The parallelization exploits data parallelism. The computational grid is partitioned with two goals in mind. Load must be balanced among the processors and overall communication cost must be minimized. The parallel execution is controlled locally on each processor. It can dynamically choose the most efficient order of execution according to the local communication status and computational load. There is no central control, which could become a bottleneck with growing numbers of processors participating in the computations.

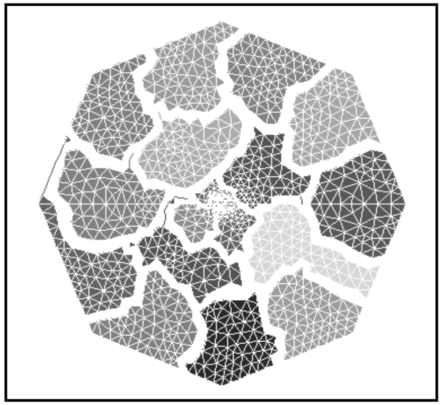
Tools

In the GRIDS project a variety of tools for the support of parallel programming has been developed. These tools can be used independently of the GRIDS system, too.

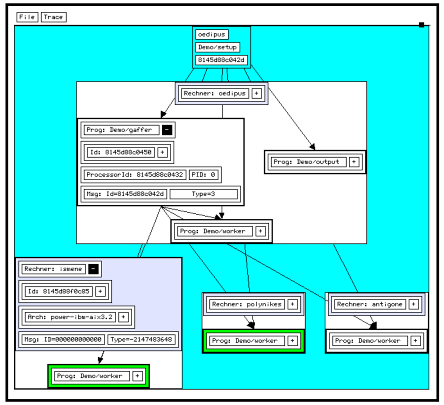
SPPL is a communication library supporting, among other features, abstract data types for messages and communication in heterogeneous workstations environments.

XPDB is a graphical debugger for the analysis of dataflow during the execution of parallel programs. The flow information is visualized hierarchically. Messages between any two processors can be displayed and manipulated in human readable format.

VISU performs the three-dimensional visualization of a partitioned computational grid.



Visualization of the partitions



XPDB execution visualization

Realization

GRIDS has been developed for parallel systems with distributed memory. This includes “real” parallel computers and workstation networks. All system dependencies are concentrated in the communication library SPPL as a basic part of the runtime system. As a consequence porting GRIDS to a new architecture requires minimal work. Application code remains untouched. The runtime system is implemented in C, writing iteration steps is done in FORTRAN.

Currently GRIDS is implemented on IBM RS/6000 workstations connected by a standard LAN. It is planned to port GRIDS to other workstation architectures, to heterogeneous environments, and to parallel systems of MIMD architecture (e.g. Intel Paragon).

References: U. Geuder, M. Härdtner, B. Wörner, R. Zink
GRIDS User’s Guide
Computer Science Report, 4/93
Computer Science Department, University of Stuttgart

Contact: Dipl.-Inform. Uwe Geuder, Dipl.-Inform. Monika Härdtner,
Dipl.-Inform. Bernhard Wörner, Dipl.-Inform. Roland Zink
IPVR, email: <FirstName.LastName>@informatik.uni-stuttgart.de