RES User's Seminar Programming models (STARSs)

Some details about the Programming Models that will be explained:

STAR superscalar (STARSs) is node level programming model based in C/Fortran + directives that nicely integrates in hybrid MPI/StarSs and provides a natural support for heterogeneity. COMPSs and SMPSs are two of the instances of StarSs. The course will focus on the implementation of COMPSs and SMPSs for MareNostrum.

COMP Superscalar (COMPSs) is a new version of GRID Superscalar which aims to easing the development of Grid. It exploits the inherent parallelism of applications when running in the Grid and has three main distinctive features with respect to its predecessor

In this sense, COMP Superscalar meets the need for programming models that simplify the development of Grid/Cluster applications: the main objective of COMP Superscalar is to keep the Grid/Cluster as transparent as possible to the programmer.

With COMP Superscalar, a sequential Java application that invokes methods of a certain granularity (tasks) is automatically converted into a parallel application whose tasks are executed in different resources of a computational Grid/Cluster. COMPSs also offers a binding to C.

SMP superscalar (SMPSs) is a programming environment focused on the ease of programming, portability and flexibility that is based on Cell superscalar (CellSs). While CellSs is tailored for the Cell/Broadband Engine processor, SMPSs is tailored for multi-cores and Symmetric Multiprocessors (SMP) in general.

SMP superscalar (SMPSs) addresses the automatic exploitation of the functional parallelism of a sequential program in multicore and SMP environments. The focus is on the portability, simplicity and flexibility of the programming model. Based on a simple annotation of the source code, a source to source compiler generates the necessary code and a runtime library exploits the existing parallelism by building at runtime a task dependency graph. The runtime takes care of scheduling the tasks and handling the associated data. Besides, a temporal locality driven task scheduling has been implemented.

The SMPSs programming environment consists of a source to source compiler and a supporting runtime library. The compiler translates C code with the aforementioned annotations into common C code with calls to the supporting runtime library. Then it compiles the resulting code using the platform C compiler.

organized by:

RES
RED ESPAÑOLA DE SUPERCOMPUTACIÓN

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA E INNOVACIÓN

BSC
Barcelona Supercomputing Center
Centro Nacional de Supercomputación