

SymGrid-Par: A Standard Skeleton-Based Framework for Computational Algebra Systems

Phil Trinder

Heriot-Watt University
P.W.Trinder@hw.ac.uk

Abstract

The SymGrid-Par framework is being developed as part of the European FP6 SCIENCE project (I3-026133) to provide a standard skeleton-based framework for parallelising large computational algebra problems in the Maple, GAP, Kant and Mupad systems.

The computational algebra community uses a number of domain specific high level languages each with specific capabilities, for example GAP specialises in computations over groups. The community are keen to develop standards, to improve interoperability between computer algebra systems (CAS), and to avoid duplicating implementation effort. Algebraic computations are challenging to parallelise as they are symbolic rather than numeric, and hence require a relatively rich set of data structures. Parallel tasks are often generated dynamically, and are of highly irregular size, e.g. varying in size by 5 orders of magnitude.

SymGrid-Par orchestrates sequential computational algebra (CA) components into a parallel, and possibly grid-enabled application. It provides a native skeleton-based interface to the CA programmer, so for example a GAP programmer might invoke a parallel GAP map function. There are both generic skeletons like map and reduce, and domain specific skeletons like orbit and transitive closure.

The skeletons are implemented by a coordination server that distributes the work to multiple instances of the sequential CAS on multiple processors; dynamically manages load distribution; and reassembles the results for return to the invoking CAS. The coordination server exploits the dynamic parallelism and load management capabilities of the Eden and GpH parallel Haskells. Invocations between SymGrid-Par components use our new standardised SCSCP protocol, currently supported by 7 CAS, and mathematical objects are represented in the standard XML-based OpenMath format.

The generic SymGrid-Par framework delivers performance comparable with, and typically better than, a specialised parallel CAS implementation like ParGAP. Many CA problems have large task granularity, and hence SymGrid-Par gives good performance on both cluster and multicore architectures. Moreover, the standardised interface can be exploited to orchestrate multiple CAS to solve problems that cannot be solved in a single CAS.

In the newly-started HPC-GAP project we are adapting the SymGrid-Par design for emerging large-scale HPC architectures. Key issues are scaling to 10^6 cores, controlling locality, and recovering from failures.

Categories and Subject Descriptors D.1.3 [Concurrent Programming]

General Terms Design, Performance, Standardization

Keywords Skeletons, Computational Algebra