

MARMOT: an MPI Analysis and Checking Tool

Bettina Krammer, Katrin Bidmon, Matthias S. Müller, Michael M. Resch

High Performance Computing Center Stuttgart
Allmandring 30, D-70550 Stuttgart, Germany
{krammer, bidmon, mueller, resch}@hlrs.de

The Message Passing Interface (MPI) is widely used to write parallel programs using message passing. MARMOT is a tool to aid in the development and debugging of MPI programs. One of its design goals is portability: by verifying that the program adheres to the MPI standard it enables the program to run on any platform in a smooth and seamless way. Another goal is scalability: the use of automatic techniques that do not need user intervention allows to debug programs running on hundreds or thousands of processors. The third goal is reproducibility: the tool contains a mechanism to detect possible race conditions and deadlocks.

MARMOT adds an additional MPI process for all tasks that cannot be handled within the context of a single MPI process, like deadlock detection. The tool uses the MPI profiling interface to intercept the MPI calls for further analysis. For example, when using `MPI_BCAST` the validity of the count, datatype, root and communicator parameters are automatically checked. MARMOT also verifies if requests are handled properly when using non-blocking send and receive or wait and test routines, e.g. if unregistered requests are used or if active requests are recycled.

Introducing race conditions in parallel programs is one of the major problems of code development, e.g. by the use of a receive call with `MPI_ANY_SOURCE` as source argument. By inspecting all calls and arguments MARMOT can detect dangerous calls and issue warnings.

MARMOT is currently being developed within the European CrossGrid project. It has been tested both with large applications, such as high energy physics or air pollution monitoring, and with benchmarks developed within this project.