

Open Source HPC Research Tools at the Institute for Scientific Computing



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Christian Bischof, Michael Burger, Alexander Hück, Tim Jammer, Jan-Patrick Lehr
December 15, 2020

General Outline This talk gives an overview of the various open source HPC-related research projects at the Institute for Scientific Computing at TU Darmstadt. For each of the projects, we present an overview of the tool and its application from a user perspective. In addition, we cover more technical details, e.g., main programming language, build system or the general development stack. We have consolidated our efforts into a single GitHub space at <https://github.com/tudasc>.

MACH MACH (MPI Assertion Checking) is a Clang/LLVM-based static analysis tool to detect if one of the recently proposed MPI-assertions can be applied to an MPI communicator. A programmer may specify these assertions, such that the MPI Library can use more optimized operations, as some of the standard's constraints are relaxed in that case, e.g., strict message ordering. MACH statically checks whether the relaxation of some of the MPI standard's guarantees will cause errors in an application. The current version supports C and C++ applications, with the scope of the analysis being limited to one translation unit. The project is available under the Apache-2.0 License.

PIRA The Performance Instrumentation Refinement Automation framework is a profiler built on top of Clang/LLVM and Score-P. It performs automatic instrumentation refinement for performance measurements of sequential and MPI-parallel applications. To reduce runtime overhead it constructs, fully automatically, filter lists for Score-P based on static source-code features and profile information. The project is available under BSD 3-clause license.

qTESLA qTESLA is a post-quantum signature scheme. Hence, it is expected to be resistant against attacks performed by a quantum-computer. Its security is based on the ring learning with errors problem. Our implementation is based on the reference implementation in C but considerably increases the performance and the useability of qTESLA. The C-library is wrapped behind the Java Native Interface (JNI). Usage is further simplified by implementing the Java Cryptographic Architecture (JCA). This qTESLA version is integrated in the CogniCrypt framework to increase availability and usability. The project is available under the MIT License.

SimAnMo The **Simulated Annealing Modeler** is a software tool to model the runtime behavior of programs depending on the size of one input parameter x . Its modeling procedure is based on runtime measurements for small values of x which result in acceptable runtimes and are so-called training points. SimAnMo supports modeling for programs with polynomial behavior, i.e., the program's runtime $t(x)$ only grows in polynomial terms with the size of the input x . In addition, SimAnMo recognizes and models and exponential growth in the runtime. SimAnMo is highly configurable and parameterizable and automatically produces PDF reports of runtime behavior. The project is available under the MIT License.

TypeART TypeART is a memory type sanitizer based on the LLVM compiler framework. In particular, TypeART is used for type correctness checks of MPI communication buffers and checkpointing libraries. In both cases, data is passed as a type-less `void` buffer, and the user is required to specify its type manually. This is error prone. To avoid a mismatch of types during runtime, TypeART instruments memory allocations (heap, stack, global) and extracts their type information to enable runtime comparisons. Detected mismatches are reported at runtime. The project is available under BSD 3-clause license.