JARA Jülich Aachen Research Alliance

```
for (i = 0; i < size; i++)
    {
        /* This will produce warnings:
         * the count is 0,
         * the types are for reduction functions.
         */
        MPI_Issend(&int_send_buf, 0, MPI_LONG_INT, right, MSG_TAG,
                   MPI_COMM_WORLD, &request);
        MPI_Recv(&int_recv_buf, 0, MPI_LONG_INT, left, MSG_TAG,
                 MPI_COMM_WORLD, &status);
        MPI_Wait(&request, &status);
        int_sum += int_recv_buf;
        int_send_buf = int_recv_buf;

        float_sum += float_recv_buf;
        float_send_buf = float_recv_buf;
    }
    printf("Incorrect result: PE%i:\tSum = %i\t%f\n", my_rank, int_sum,
           float_sum);
    /* Now correctly: */
    /* Compute global sum. */
    int_sum = 0;
    float_sum = 0;
    int_send_buf = my_rank;
    float_send_buf = (float) my_rank;
    for (i = 0; i < size; i++)
    {
        MPI_Issend(&int_send_buf, 1, sendtype, right, MSG_TAG,
                   MPI_COMM_WORLD, &request);
        MPI_Recv(&int_recv_buf, 1, recvtype, left, MSG_TAG,
                 MPI_COMM_WORLD, &status);
        MPI_Wait(&request, &status);
        int_sum += int_recv_buf;
        int_send_buf = int_recv_buf;
        float_sum += float_recv_buf;
        float_send_buf = float_recv_buf;
    }
```

Do you spot the error? If not, use MUST

MUST - Scalable Runtime Correctness Checking

AN INITIATIVE OF

RWTH AACHEN UNIVERSITY

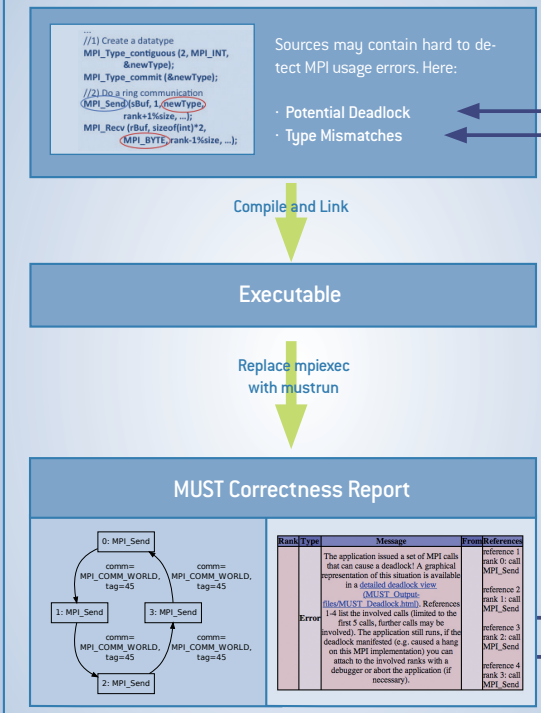
JÜLICH FORSCHUNGSZENTRUM

MUST

Runtime Error Detection for Parallel Applications

The Marmot Umpire Scalable Tool (MUST) is a runtime error detection tool that automatically detects non-standard compliant use of MPI. MUST attaches to the target application and intercepts its MPI calls. It checks for local and non-local correctness errors, such as errors in point-to-point type matching or deadlocks. It unites the features of Marmot and Umpire while providing drastically increased scalability at the same time and is available as open source.

Runtime analysis with MUST



MUST detects:

- Invalid arguments
- Resource usage errors and leaks
- Actual and potential deadlocks
- Datatype matching errors
- Overlaps in communication buffers

Currently MUST provides a rich set of correctness checks where checks are available for large scale runs. Especially, we present a first scalable runtime deadlock detection during ISC. For that we provide an open source base infrastructure called the Generic Tool Infrastructure (GTI). It eases many hurdles in tool development and allows fast and efficient specification of scalable tools. Furthermore GTI offers modes to finish the runtime analysis while the target application crashes. We also provide GTI as open source.

Developed by:

ZIH Center for Information Services & High Performance Computing

RWTH AACHEN UNIVERSITY

Los Alamos NATIONAL LABORATORY

Lawrence Livermore National Laboratory

Contact: terboven@itc.rwth-aachen.de · http://www.itc.rwth-aachen.de/must