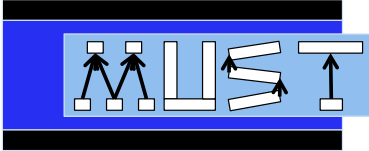


MUST

Runtime Error Detection for MPI 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 is based on the existing tools Marmot and Umpire and unites their features while providing drastically increased scalability at the same time. We provide this easy to use tool as open source.

MUST detects:

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

The current MUST version provides a rich set of correctness checks where most checks are available for large scale runs. Even for features such as deadlock detection we provide scalability for several hundred tasks depending on the use-case. For that we provide a 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.

Get your demo during SC12
at the ZIH booth!

Booth #4036, Hall 2

Application Source Code

```
...  
//1) Create a datatype  
MPI_Type_contiguous(2, MPI_INT,  
    &newType);  
MPI_Type_commit(&newType);  
//2) Do a ring communication  
MPI_Send(sBuf, 1, newType,  
    rank+1*size, ...);  
MPI_Recv(rBuf, sizeof(int)*2,  
    MPI_BYTE, rank-1*size, ...);  
...
```

Sources may contain hard to detect MPI usage errors. Here:

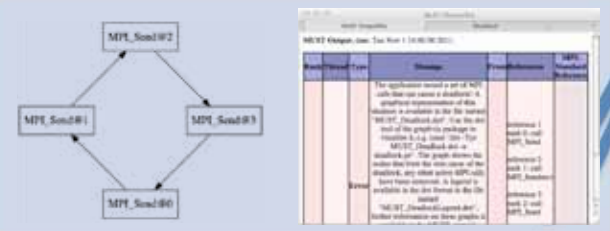
- Potential Deadlock
- Type Mismatches

Compile & Link

Executable

Replace *mpiexec*
with *mustrun*

MUST Correctness Report



Partners:

