

Exposing MPI Objects for Debugging

Motivation

- Debugging MPI applications is difficult as developers cannot inspect the state of the MPI runtime
- The MPI Tools Working Group has proposed a standard interface between the debugger and MPI runtime [1]
- With this interface, debuggers can easily present MPI state to the developer

Contributions

- Support for MPI handle introspection in the *TotalView* debugger
- A reference introspection implementation in *Open MPI*
- A demonstration of simplified MPI debugging

Debugger access to MPI state

- MPI implementation provides functions for *introspection* as a library
- Introspection functions extract data from MPI handles
- Debugger and library interact through mutual callback functions [2]

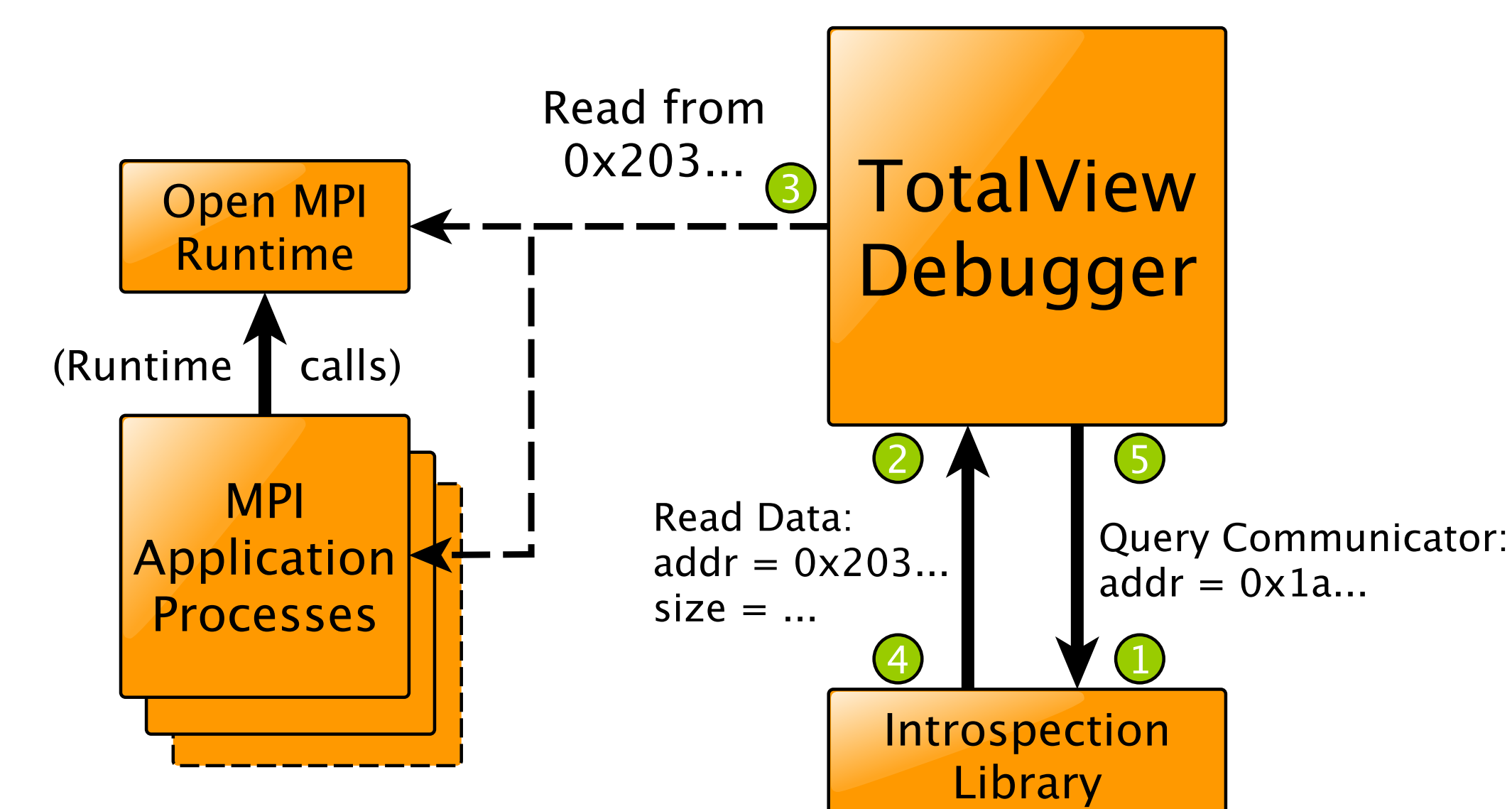


Figure 1 : Example use case to fetch communicator information.

- Debugger queries a communicator
- Library requests raw data from process
- Debugger extracts raw data from process
- Library parses raw data and returns standardized communicator information
- Debugger presents data to developer

Complex MPI Debugging Session with a Debugger

Attach debugger to application

Browse internal Handle data structure

Name of communicator

Rank

Flags need to be decoded manually

Continue to internal process group data structure

Locate first process in communicator

And so forth...

Here we find name of host

Figure 2 : Best case scenario for existing MPI debugging. Developer must understand internal data structures. For closed source MPI implementations, even this is not possible.

Simplified Debugging with MPI Handle Introspection in TotalView

Only show relevant data, not raw data structures

```

d1.<> .mpidbg
Loaded MPI support library /g/g90/laustbn/local/lib/
openmpi/libomp_dbg_mphandles.so :
Open MPI handle interpretation support for parallel
debuggers compiled on Sep 5 2014

Finished loading MPI introspection support.

d1.<> dfocus p2
p2.<
p2.<> .mpidbgdump
Name                Handle
MPL_COMM_WORLD      0x6028a0
MPL_COMM_SELF       0x2aaaa01aa00
MPL_COMM_PARENT     0x2aaaa01a9e0
MPL_COMM_NULL       0x2aaaa01a3e0

p2.<> .mpidbgquery basic 0x6028a0
Querying communicator 0x6028a0 in process 0x4878780
Communicator: MPL_COMM_WORLD
Rank: 0
Size: 4
Flag                Value
MPIDBG_COMM_INFO_PREDEFINED  True
MPIDBG_COMM_INFO_CARTESIAN   False
MPIDBG_COMM_INFO_GRAPH       False
MPIDBG_COMM_INFO_TOPO_REORDERED False
MPIDBG_COMM_INFO_INTERCOMM   False
{...}
Query was successful
    
```

Implementation defined structures such as bit fields and types are automatically decoded

Figure 3 : Debugging with introspection support through the TotalView command line. TotalView displays the status of a communicator.

- MPI implementation traverses data structures – assisted by the Debugger
- Developer sees MPI API level information

Future Work

- Integration into TotalView's graphical user interface
- Validation against other MPI implementations
- Visualization of MPI communication and processes
- Querying MPI handle state from the debugger; filtering and showing handles matching certain criteria

Acknowledgements

The authors would like to thank Adam Moody at Lawrence Livermore National Laboratory for useful insights on MPI debugging.

This work (LLNL-POST-658417) was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, and co-funded by the European Artemis PaPP Project nr. 295440 and COPCAMS project nr. 332913.

References

- MPI Tools Working Group. <https://svn.mpi-forum.org/trac/mpi-forum-web/wiki/MPI3Tools>, October 2014.
- James Cownie and William Gropp. A Standard Interface for Debugger Access to Message Queue Information in MPI. In *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, pages 51–58. Springer, 1999.

Contact Information

Laust Brock-Nannestad
 Technical University of Denmark
 Email: laub@dtu.dk
 Phone: +45 4525 9223

<http://www.compute.dtu.dk/~laub/>