A Clock-Based Approach to Detect Memory Consistency Errors in MPI One-Sided Communication

Michael Shell
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250
Email: http://www.michaelshell.org/contact.html

Homer Simpson
Twentieth Century Fox
Springfield, USA
Email: homer@thesimpsons.com

James Kirk
and Montgomery Scott
Starfleet Academy
San Francisco, California 96678-2391
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—The abstract goes here.**

1. Introduction

2. Memory Consistency Errors

- What is a memory consistency error?
- Showing several examples of memory consistency errors

3. MC-Checker

There are three modules in MC-Checker:

- ST-Analyzer
- Profiler
- DN-Analyzer

4. Our Clock-Based Approach

There are three modules in our clock-based approach: Analyzer, Profiler and Detector.

1) Analyzer is implemented by
   a) static analysis
   b) text processing
2) Profiler is implemented by MPI Profiling as follows:
   a) For MPI_Win_fence, MPI_Barrier: using PMPI_Allgather to update vector clocks
   b) For MPI_Win_post/complete: using PMPI_Send to update vector clocks
   c) For MPI_Win_start/wait: using PMPI_Recv to update vector clocks
   d) For MPI_Send: using PMPI_Pack to update vector clocks
   e) For MPI_Recv: using PMPI_Unpack to update vector clocks
3) Detector
   a) Matching synchronization calls
   b) Detecting conflicting operations within an epoch
   c) Detecting conflicting operations across processes

5. Evaluation

- Correctness
- Slowdown
- Memory Usage

6. Discussion

7. Related Work

The following are some debugging tools or techniques to detect bugs in MPI one-sided communication:

- Marmot [1]
- [2]
- "Mirror Memory" [3]
- SyncChecker [4]
- MC-Checker [5]
- Nasty-MPI [6], [7]

8. Conclusions and Future Work

- The size of vector clocks

Acknowledgments

The authors would like to thank...

References

[1] B. Krammer and M. M. Resch, "Correctness checking of mpi one-sided communication using marmot," in *European Parallel Virtual Machine/Message Passing Interface Users Group Meeting*. Springer, 2006, pp. 105–114.

[2] S. Pervez, G. Gopalakrishnan, R. M. Kirby, R. Thakur, and W. Gropp, "Formal verification of programs that use mpi one-sided communication," in *European Parallel Virtual Machine/Message Passing Interface Users Group Meeting*. Springer, 2006, pp. 30–39.

[3] M.-Y. Park and S.-H. Chung, "Detecting race conditions in one-sided communication of mpi programs," in *Computer and Information Science, 2009. ICIS 2009. Eighth IEEE/ACIS International Conference on*. IEEE, 2009, pp. 867–872.

[4] Z. Chen, X. Li, J.-Y. Chen, H. Zhong, and F. Qin, "Syncchecker: Detecting synchronization errors between mpi applications and libraries," in *Parallel & Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International*. IEEE, 2012, pp. 342–353.

[5] Z. Chen, J. Dinan, Z. Tang, P. Balaji, H. Zhong, J. Wei, T. Huang, and F. Qin, "Mc-checker: Detecting memory consistency errors in mpi one-sided applications," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Press, 2014, pp. 499–510.

[6] R. Kowalewski and K. Fürlinger, "Nasty-mpi: Debugging synchronization errors in mpi-3 one-sided applications," in *European Conference on Parallel Processing*. Springer, 2016, pp. 51–62.

[7] ——, "Debugging latent synchronization errors in mpi-3 one-sided communication," pp. 83–96, 2017.