



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Performance Analysis of and Tool Support for Transactional Memory on BG/Q

M. Schindewolf

December 9, 2011

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Technical Report – Performance Analysis of and Tool Support for Transactional Memory on BG/Q

Martin Schindewolf

Martin Schindewolf worked during his internship at the Lawrence Livermore National Laboratory (LLNL) under the guidance of Martin Schulz at the Computer Science Group of the Center for Applied Scientific Computing. We studied the performance of the TM subsystem of BG/Q as well as researched the possibilities for tool support for TM.

To study the performance, we run CLOMP-TM. CLOMP-TM is a benchmark designed for the purpose to quantify the overhead of OpenMP and compare different synchronization primitives. To advance CLOMP-TM, we added Message Passing Interface (MPI) routines for a hybrid parallelization. This enables to run multiple MPI tasks, each running OpenMP, on one node. With these enhancements, a beneficial MPI task to OpenMP thread ratio is determined. Further, the synchronization primitives are ranked as a function of the application characteristics. To demonstrate the usefulness of these results, we investigate a real Monte Carlo simulation called Monte Carlo Benchmark (MCB). Applying the lessons learned yields the best task to thread ratio. Further, we were able to tune the synchronization by transactivating the MCB.

Further, we develop tools that capture the performance of the TM run time system and present it to the application’s developer. The performance of the TM run time system relies on the built-in statistics. These tools use the Blue Gene Performance Monitoring (BGPM) interface to correlate the statistics from the TM run time system with performance counter values. This combination provides detailed insights in the run time behavior of the application and enables to track down the cause of degraded performance. Further, one tool has been implemented that separates the performance counters in three categories: *Successful Speculation*, *Unsuccessful Speculation* and *No Speculation*.

All of the tools are crafted around IBM’s *xlc* compiler for C and C++ and have been run and tested on a Q32 early access system.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344 (LLNL-TR-519833).