Level 1 Parallel RTN-BLAS: Implementation and Efficiency Analysis

Chemseddine Chohra, Philippe Langlois, David Parello

Level 1 Parallel RTN-BLAS: Implementation and Efficiency Analysis

Chemseddine Chohra,
Philippe Langlois and David Parello

Univ. Perpignan Via Domitia, Digits, Architectures et Logiciels Informatiques, F-66860, Perpignan. Univ. Montpellier II, Laboratoire d'Informatique Robotique et de Microélectronique de Montpellier, UMR 5506, F-34095, Montpellier. CNRS, Laboratoire d'Informatique Robotique et de Microélectronique de Montpellier, UMR 5506, F-34095, Montpellier.
chemseddine.chohra@univ-perp.fr

Keywords: Floating point arithmetic, numerical reproducibility, Round-To-Nearest BLAS, parallelism, summation algorithms.

Modern high performance computation (HPC) performs a huge amount of floating point operations on massively multi-threaded systems. Those systems interleave operations and include both dynamic scheduling and non-deterministic reductions that prevent numerical reproducibility, *i.e.* getting identical results from multiple runs, even on one given machine. Floating point addition is non-associative and the results depend on the computation order. Of course, numerical reproducibility is important to debug, check the correctness of programs and validate the results. Some solutions have been proposed like parallel tree scheme [1] or new Demmel and Nguyen's reproducible sums [2]. Reproducibility is not equivalent to accuracy: a reproducible result may be far away from the exact result. Another way to guarantee the numerical reproducibility is to calculate the correctly rounded value of the exact result, *i.e.* extending the IEEE-754 rounding properties to larger computing sequences. When such computation is possible, it is certainly more costly. But is it unacceptable in practice?
We are motivated by round-to-nearest parallel BLAS. We can implement such RTN-BLAS thanks to recent algorithms that compute correctly rounded sums. This work is a first step for the level 1 of the

BLAS routines. We study the efficiency of computing parallel RTN-sums compared to reproducible or classic ones – MKL for instance. We focus on HybridSum and OnlineExact, two algorithms that smooth the over-cost effect of the condition number for large sums [3,4]. We start with sequential implementations: we describe and analyze some hand-made optimizations to benefit from instruction level parallelism, pipelining and to reduce the memory latency. The optimized over-cost is at least 25% reduced in the sequential case. Then we propose parallel RTN versions of these algorithms for shared memory systems. We analyze the efficiency of OpenMP implementations. We exhibit both good scaling properties and less memory effect limitations than existing solutions. These preliminary results justify to continue towards the next levels of parallel RTN-BLAS.

References:

[**1**] O. Villa, D. G. Chavarría-Miranda, V. Gurumoorthi, A. Márquez, and S. Krishnamoorthy. Effects of floating-point non-associativity on numerical computations on massively multithreaded systems. In *CUG Proceedings*, (2009), pp. 1–11.

[**2**] James Demmel and Hong Diep Nguyen, Fast Reproducible Floating-Point Summation. In *21st IEEE Symposium on Computer Arithmetic, Austin, TX, USA, April 7-10*, (2013), pp. 163—172.

[**3**] Yong-Kang Zhu and Wayne. B. Hayes, Correct rounding and a hybrid approach to exact floating-point summation. *SIAM J. Sci. Comput.*, (2009), Vol. 31, No. 4, pp. 2981–3001.

[**4**] Yong-Kang Zhu and Wayne. B. Hayes, Algorithm 908: Online exact summation of floating-point streams. *ACM Trans. Math. Software*, (2010), 37:1–37:13.