LITERATURE REVIEW: Accurately computing large floating-point numbers using parallel computing

Tanvir Kaykobad

School of Computer Science

Carleton University

Ottawa, Canada K1S 5B6

TanvirKaykobad@cmail.carleton.ca

October 15, 2017

1 Introduction

It is known that for accurately summing up a very large number of small and large floating point numbers one has to use snowball effect to obtain larger numbers from small ones. We like to pursue this idea and extend it in parallel computing. For that we study the existing literature on how to compute the exact sum of n floating numbers parallely.

2 Literature Review

Give an overview of the relevant literature. Cite all relevant papers, like [2], [12], [3], [10], [5], [1], and [4]. Outline for each paper the relevant results in relation to your project. Make sure that you don't just list all relevant papers in random order. Devise a scheme to group papers by subject. The goal is to present to the reader the state-of-the-art in the field selected for your project.

Demmel and Nguyen [7] used Rumps algorithm for floating point summation that is reproducible independent of the order of summation that may be different with the dynamic scheduling of parallel computing resources and floating point nonassociativity. Their absolute error bound is 2-28 times macheps, and requires constant amount of extra memory usage.Demmel and Hida [6] analysed several algorithms and showed that if a wider accumulator of F bits is used to sum n floating point numbers each of at most f bits, and if sum is carried out in descending order of exponents then an error of at most 1.5 times the least significant bit can occur provided that number of summands does not exceed $2^{(F-f)}$. Rump, Ogita and Oishi [13] showed that their algorithm results in a value nearest to the true sum. In 2013 the authors [8] have shown how to use tree reduced parallelism to compute sum by using parallel associative reduction, iterative refinement and conservative early detection to obtain an algorithm of order log n. Neal [11] presented two algorithms in one of which a small superaccumulator with 67 64-bit chunks each with 32-bit overlap with the next chunk was used to allow carry propagation to be done infrequently. Kai and wang [9] have shown that computing ensuring minimum error when n numbers can be both positive and negative is NP-hard. However their algorithm can sum with no more than 2(ceiling (log (n-1)+1)*E $2\lceil \log{(n-1)} + 1 \rceil * \epsilon$, where ϵ is the worst case minimum error over all possible orders.

References

[1] Y. Chen, F. Dehne, T. Eavis, D. Green, A. Rau-Chaplin, and E. Sithirasenan. cgmO-LAP: Efficient parallel generation and querying of terabyte size ROLAP data cubes. In *Proc. 22nd Int. Conf. on Data Engineering (ICDE)*, pages 164–164. IEEE Comp. Soc. Dig. Library, 2006.

[2] F. Dehne, T. Eavis, and B. Liang. Compressing data cubes in parallel OLAP systems. *Data Science Journal*, 6:S184–S197, 2007.

[3] F. Dehne, T. Eavis, and A. Rau-Chaplin. The cgmCUBE project: Optimizing parallel data cube generation for ROLAP. *Distributed and Parallel Databases*, 19(1):29–62, 2006.

[4] F. Dehne, M. Fellows, M. Langston, F. Rosamond, and K. Stevens. An $o(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem. In *Proc. 11th Int. Computing and Combinatorics Conf. (COCOON)*, pages 859–869. Springer LNCS 3595, 2005.

[5] F. Dehne, M. Langston, X. Luo, S. Pitre, P. Shaw, and Y. Zhang. The cluster editing problem: Implementations and experiments. In *Proc. Int. Workshop on Parameterized and Exact Computation (IWPEC)*, pages 13–24. Springer LNCS 4169, 2006.

[6] J. Demmel and Y. Hida. Accurate and efficient floating point summation. *SIAM Journal on Scienti c Computing*, 25(4):1214–1248, 2004.

[7] J. Demmel and H. D. Nguyen. Parallel reproducible summation. *IEEE TC*, 64(7):2060–2070, July 2015.

[8] E. Kadric, P. Gurniak, and A. DeHon. Accurate parallel floating-point accumulation. *21st IEEE Symp. on Computer Arithmetic (ARITH)*, 31(1):153–162, April 2013.

[9] M.-Y. Kao and J. Wang. Linear-time approximation algorithms for computing numerical summation with provably small errors. *SISC*, 29(5):1568–1576, 2000.

[10] M. Lawrence, F. Dehne, and A. Rau-Chaplin. Implementing OLAP query fragment aggregation and recombination for the OLAP enabled grid. In *Proc. International Parallel and Distributed Processing Symposium (IPDPS), High-Performance Grid Computing Workshop*, pages 1–8. IEEE Comp. Soc. Dig. Library, 2007.

[11] R. M. Neal. Fast exact summation using small and large superaccumulators. *arXiv ePrint*, abs/1505.05571:153–162, 2015.

[12] S. Pitre, F. Dehne, A. Chan, J. Cheetham, A. Duong, A. Emili, M. Gebbia, J. Greenblatt, M. Jessulat, N. Krogan, X. Luo, and A. Golshani. PIPE: a protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs. *BMC Bioinformatics*, 7:365 (15 pages), 2006, available via PubMed at http://www.biomedcentral.com/pubmed/16872538.

[13] S. M. Rump, T. Ogita, and S. Oishi. Accurate floating-point summation part i: Faithful rounding. *SIAM Journal on Scienti c Computing*, 31(1):189–224, 2008.