Wolfgang Lehner, Dirk Habich, Till Kolditz

Teaching In-Memory Database Systems the Detection of Hardware Errors

Erstveröffentlichung in / First published in:

2018 IEEE 34th International Conference on Data Engineering. Paris, 16.-19.04.2018. IEEE, S.1663. ISBN 978-1-5386-5520-7.

DOI: http://dx.doi.org/10.1109/ICDE.2018.00203

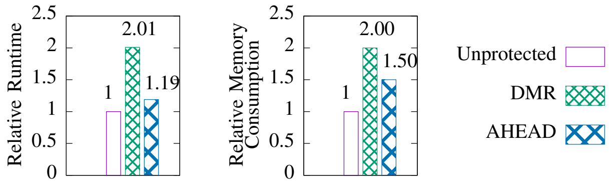
Final edited form was published in "2018 IEEE 34th International Conference on Data Engineering. Paris 2018", S.1663 , ISBN 978-1-5386-5520-7
http://dx.doi.org/10.1109/ICDE.2018.00203

Teaching In-Memory Database Systems the Detection of Hardware Errors

Wolfgang Lehner, Dirk Habich, Till Kolditz
Dresden Database Systems Group
Technische Universität Dresden, Germany
{firstname.lastname}@tu-dresden.de

The key objective of database systems is to reliably manage data, whereby high query throughput and low query latency are core requirements. To satisfy these requirements, database systems constantly adapt to novel hardware features on the one hand. On the other hand, we have already known for a long time that hardware components are not perfect and soft errors in terms of single bit flips happen all the time. Today, hardware-based protection is the common approach to mitigate these single bit flips. However, recent studies have shown that future hardware is becoming less and less reliable and that multi-bit flips may prevail single bit flips. For example, repeatedly accessing one memory cell in DRAM modules causes bit flips in physically-adjacent memory cells, whereby one to four bits flips per 64-bit word have been discovered [1]. Furthermore, emerging non-volatile memory technologies like PCM exhibit even more reliability issues [2], [3]. For instance, heat produced by writing one PCM cell can alter the value stored in many nearby cells (e.g., up to 11 cells in a 64-byte block [4]). Additionally, hardware aging effects will lead to changing bit flip rates at run-time [5]. Unfortunately, scaling hardware-based protection techniques to cover changing multi-bit flips is possible, but this introduces large performance, chip area, and power overheads, which will become unaffordable in the future [5], [6], [7].

Consequently, this shift also affects database systems, because data as well as query processing have to be protected in software accordingly to further guarantee a reliable data management on future unreliable hardware. Generally, any undetected bit flip destroys the reliability objective in form of false negatives (missing tuples), false positives (tuples with invalid predicates), or inaccurate aggregates in a silent way. To tackle theses issues from a database perspective, we developed *AHEAD* [8], a novel adaptable and on-the-fly hardware error detection approach for in-memory column stores. AHEAD provides configurable error detection in an end-to-end fashion by using an arithmetic error coding technique which allows query processing to completely work on encoded data. This enables on-the-fly error detection during query processing (i) which modifies data stored in memory or transferred on an interconnect, and (ii) which are induced during computations. Compared to state-of-the-art protection approaches like dual modular redundancy, AHEAD reduces the overhead dramatically. For instance, DMR protection requires twice as much memory capacity compared to an unprotected setting, since data must be kept in two different memory locations. Furthermore, every query is redundantly executed with an additional voting at the end resulting in a performance overhead higher than 100%. Using *AHEAD*, we observed an average performance overhead over all SSB queries of 19% compared to an unprotected setting as illustrated in Fig.1.



(a) Runtime (b) Storage

Fig. 1: Comparing the unprotected in-memory database concept with protected concepts of double modular redundancy (DMR) and our *AHEAD* approach using the SSB benchmark.

In our talk, we will present the current state of our research direction of teaching in-memory database systems the detection of hardware errors. We will stress the fact, that our favored error coding approach is *orthogonal* to other coding domains like compression or encryption, which allows a free combination of different schemes of the underlying data. Finally, we will also outline our future work in this context.

Acknowledgments

This work is funded by DFG within the Cluster of Excellence *Center for Advancing Electronics Dresden (cfaed).*

REFERENCES

[1] Y. Kim, R. Daly, J. Kim, C. Fallin, J. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu, "Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors," in *ISCA*, 2014.

[2] S. Mittal, "A survey of soft-error mitigation techniques for non-volatile memories," *Computers*, vol. 6, no. 1, p. 8, 2017.

[3] O. Mutlu, "The rowhammer problem and other issues we may face as memory becomes denser," in *DATE*, 2017, pp. 1116–1121.

[4] L. Jiang, Y. Zhang, and J. Yang, "Mitigating write disturbance in super-dense phase change memories," in *DSN*, 2014, pp. 216–227.

[5] S. Rehman, M. Shafique, and J. Henkel, *Reliable Software for Unreliable Hardware - A Cross Layer Perspective*. Springer, 2016.

[6] J. Henkel, L. Bauer, N. Dutt, P. Gupta, S. R. Nassif, M. Shafique, M. B. Tahoori, and N. Wehn, "Reliable on-chip systems in the nano-era: lessons learnt and future trends," in *DAC*, 2013, pp. 99:1–99:10.

[7] M. Shafique *et al.*, "Multi-layer software reliability for unreliable hardware," *it - Information Technology*, vol. 57, no. 3, pp. 170–180, 2015.

[8] T. Kolditz, D. Habich, W. Lehner, M. Werner, and S. de Bruijn, "Ahead: Adaptable data hardening for on-the-fly hardware error detection during database query processing," in *SIGMOD*, 2018.