Evolution of Memory Reliability Techniques

Ph.D. Candidacy Exam

Evgeny Manzhosov
Department of Computer Science, Columbia University
evgeny@cs.columbia.edu

The papers cover the topic of system reliability with an emphasis on the main memory subsystem. I begin by briefly covering how the issue of reliability was seen by the in-industry and the community: the early projections, the doubts, and long-awaited discoveries. Then, I show how memory reliability techniques evolved during the past 20+ years in the context of modern memories (i.e., DRAMs) from simple Single Error Correcting codes to device-failure correcting codes. Moreover, I will show a recent trend in memory architecture where built-in ECC becomes de facto a standard feature of the device. At last, I explore system reliability beyond the main memory. This part focuses on techniques that allow us to measure the degree of reliability in computing systems and methods enabling timely fault detection at various levels, i.e., software and/or hardware, at low cost.

1 Reliability Trends

1. **Modeling the effect of technology trends on the soft error rate of combinatorial logic** (2002)

   Models the trend of the soft error rates of digital logic and memory for various tech nodes projecting reliability to become a serious concern in the future.

2. **The impact of technology scaling on lifetime reliability** (2004)

   Shows impact of scaling on lifetime reliability of CPU while taking the workload into account.

3. **Reliability: Fallacy or Reality?** (2007)

   Do we really need to spend money on reliability?

4. **Silent Data Corruptions at Scale** (2021)

   Shows a real case of SDCs in modern hardware.

2 Mitigations

2.1 Microarchitectural for DRAM

5. **A White Paper on the Benefits of Chipkill-Correct ECC for PC Server Main Memory** (1997)

   Proposes to use ChipKill-like level of reliability.

6. **Virtualized and Flexible ECC for Main Memory** (2010)

   Proposes an ECC scheme where redundant information is virtualized in main memory.

7. **LOT-ECC: Localized and tiered reliability mechanisms for commodity memory systems** (2012)

   Multi-tier ECC scheme.

8. **Low-power, Low-storage-overhead Chipkill Correct via Multi-line Error Correction (MULTI ECC)** (2014)

   Separates detection and correction to guarantee Chipkill (use of checksums for position, and erasure codes for correction).

9. **Bamboo ECC: Strong, safe, and flexible codes for reliable computer memory** (2015)

   Treat DRAM transactions as additional space – make vertical ECCs and increase the protection strength.

10. **Frugal ECC: efficient and versatile memory error protection through fine-grained compression** (2015)

    Memory compression is used to gain storage for ECC.

11. **XED: Exposing On-Die Error Detection Information for Strong Memory Reliability** (2016)

    Exposes IECC to memory controller turning it into a system microarchitectural feature.

12. **Defect analysis and cost-effective resilience architecture for future DRAM devices** (2017)

    Advocates for the wide use of IECC.

13. **DUO: Exposing On-Chip Redundancy to Rank-Level ECC for High Reliability** (2018)

    IECC is here to stay, so what is the better use of it? The case for treating redundancy budgets as monolithic space rather than layers.

14. **Dvé: Improving DRAM Reliability and Performance On-Demand via Coherent Replication** (2021)

    Data-replication-based error correction.

2.2 Co-design: DRAM reliability and Security

15. **Combining tags with error codes** (1983)

    Gumpertz's paper on using erasure codes to store metadata tags.

16. **IVEC: off-chip memory integrity protection for both security and reliability** (2010)

    Use of Merkle-Trees integrity scheme for security and reliability w/ non-ECC DIMMs.

17. **SYNERGY: Rethinking Secure-Memory Design for Error-Correcting Memories** (2018)

    Co-designs for Security and reliability by using MACs for error detection (placed in ECC DRAM), and multi-line parity to correct the errors.

2.3 Compiler-assisted

18. **Shoestring: Probabilistic Soft Error Reliability on the Cheap** (2010)

    Selective instruction duplication to cover the faults not covered by symptom-based detection.

2.4 Architectural

19. **Argus: Low-Cost, Comprehensive Error Detection in Simple Cores** (2007)

    Observes that by checking the correctness of four fundamental tasks of compute (control flow, dataflow, computation, and memory access) is enough to guarantee reliable execution.

20. **ReStore: Symptom Based Soft Error Detection in Microprocessors** (2005)

    A novel technique to detect transient faults in CPU pipeline via "symptoms" that execution went wrong: cache and TLB misses, branch misprediction, hardware exceptions, etc.

3 Reliability Analysis and Countermeasures

3.1 Studies and Methodologies

21. **Memory Errors in Modern Systems: The Good, The Bad, and The Ugly** (2015)

Study shows that counting errors is misleading – count faults! Highlights the need for a very strong memory reliability schemes Almost 20 years later since the IBM ChipKill paper was published.

22. **A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor** (2003)

    A first paper to propose method to measure the impact of HW structures on the reliability of program execution.

23. **Eliminating microarchitectural dependency from Architectural Vulnerability** (2009)

    Allows analyzing vulnerability of a program to soft errors and shows that one can improve reliability at the program level.

24. **Using Hardware Vulnerability Factors to Enhance AVF Analysis** (2010)

    This work isolates the contribution of hardware structures to AVF. Moreover, it redefines AVF as a combination of PVF and HVF.

25. **Demystifying the System Vulnerability Stack: Transient Fault Effects Across the Layers** (2021)

    It is a mistake to separate software and hardware layers for reliability purposes.

26. **Minotaur: Adapting Software Testing Techniques for Hardware Errors** (2019)

    A toolkit to improve the analysis of software vulnerability to hardware errors by leveraging concepts from software testing.

27. **Understanding the propagation of hard errors to software and implications for resilient system design** (2008)

    This work characterizes how hard faults in various microarchitectural structures propagate through the application and the os allowing for low cost symptom-based detection.

28. **Online Estimation of Architectural Vulnerability Factor for Soft Errors** (2008)

    First paper to propose a hardware solution for estimating AVF at runtime, associated challenges, and sources of inaccuracy.