Retrospective: RAIDR: Retention-Aware Intelligent DRAM Refresh

Onur Mutlu
ETH Zürich

Abstract—Dynamic Random Access Memory (DRAM) is the prevalent memory technology used to build main memory systems of almost all computers. A fundamental shortcoming of DRAM is the need to refresh memory cells to keep stored data intact. DRAM refresh consumes energy and degrades performance. It is also a technology scaling challenge as its negative effects become worse as DRAM cell size reduces and DRAM chip capacity increases.

Our ISCA 2012 paper, RAIDR [1], examines the DRAM refresh problem from a computational perspective. It rigorously examined the questions of 1) how to perform accurate DRAM data retention time profiling, 2) how to overcome potential hurdles that stand in the way of obtaining accurate minimum data retention times, 3) how to reliably get rid of unnecessary refresh operations.

III. BUILDING ON RAIDR AND MAKING IT WORK
We believe RAIDR enabled a refreshing approach to DRAM refresh. Its largest contribution could be the works it has inspired that rigorously examined the questions of 1) how to perform accurate DRAM data retention time profiling, 2) how to overcome potential hurdles that stand in the way of obtaining accurate minimum data retention times, 3) how to reliably get rid of unnecessary refresh operations.

Apart from the new technique it introduced, we believe the RAIDR paper made two other major contributions that have enabled a large number of future works and new ideas. First, it provided an empirical scaling analysis that clearly demonstrated the importance of the DRAM refresh problem in modern systems: if nothing is done about it, DRAM refresh would waste almost half of the throughput and half of the energy of a high-capacity 64-Gb DRAM chip! This analytical prediction encouraged more works in the topic area. Second, it demonstrated a methodical way of exploiting cell-level heterogeneous data retention times at the system (e.g., memory controller) level: if data retention times of DRAM rows are accurately known, the system can use them to optimize DRAM refresh and get rid of most refresh operations. This demonstration enabled other works to develop 1) methods for accurately determining DRAM data retention time profiling and 2) other system-level approaches to optimize DRAM behavior using data retention time information.

II. CONTRIBUTIONS AND IMPACT OF RAIDR
RAIDR is the first work to propose a low-cost memory controller technique that reduces refresh operations by exploiting variation in data retention times across DRAM rows. Its appeal comes from its simplicity and low cost, enabled by the careful use of Bloom filters [15]. Exploiting the DRAM data retention time distribution [16], RAIDR can eliminate a very large fraction (e.g., ~75% or more) of refresh operations with very small hardware cost at the memory controller.
methods to reduce DRAM refresh’s negative impact on performance & energy (e.g., [41, 59–67]). Our HPCA 2014 paper [59] developed a more refined projection of the effect of DRAM refresh as technology scales. AVATAR, in DSN 2015 [26] and REAPER in ISCA 2017 [30] enabled more practical ways of exploiting heterogeneous retention times in the presence of VRT. Our recent work [66] shows that with a more flexible DRAM interface that gives some autonomy to DRAM chips, RAIDR can be more efficiently implemented inside the DRAM chip.

IV. SUMMARY AND FUTURE OUTLOOK

RAIDR is a nice example of how enthusiastic support from industry can foster new ideas that can open up many new analyses and other ideas. We were inspired by our deep technical discussions with especially Samsung and Intel, along with prior works that described DRAM technology scaling challenges (e.g., [3]) and that developed promising solutions (e.g., [66, 69]). Engineers from Samsung and Intel later wrote an insightful paper [34] on DRAM scaling challenges, which described refresh as a key problem and advocated a controller-DRAM co-design approach as we had been advocating [1, 4]. RAIDR was also a nice example of how teaching & research smoothly feed each other: much of the research was done as part of a group project in the Parallel Computer Architecture class I taught at CMU in Fall 2011.

Looking forward, DRAM technology scaling is getting worse and data retention will continue to be an important issue [34, 70]. The negative effects of DRAM refresh will be (and are being) exacerbated by other technology scaling issues like RowHammer [35] that require new solutions [41, 44, 71]. We believe there are a lot more new ideas and techniques to develop to minimize the impact of refresh on computing systems.

REFERENCES