technische universität dortmund

computer science 12

Bachelor/Master Thesis

Benchmark suit for evaluating wear levelling on specific memory regions

Nils Hölscher
Prof. Dr. Jian-Jia Chen
Otto-Hahn Str. 16
Technische Universität Dortmund
Email: nils.hoelscher@tu-dortmund.de
10.03.2022

The commercial availability of non-volatile memory (NVM) as byte-addressable random-access memory (RAM) yields new design principles in the system software and the application level. FRAM is a prominent example of a low power NVM, which can be found as on-chip RAM on some microcontrollers. However all NVM solutions are write destructive and some are also read destructive. For prolonging the life time of an NVM memory cell current research focuses on methods for wear levelling.

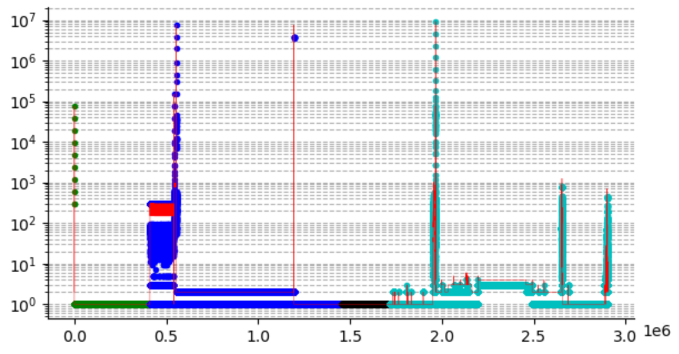

Figure 1: Number of Bit-flips for a Dijkstra run. (Red=#Accesses, Green=#Flips in data, Blue=#Flips in BSS, Black=#Flips in Heap, Magenta=#Flips in Stack).

Wear levelling describes methods for evenly distributing Bit-flips on the whole memory region. Figure 1 shows the number of Bit-flips to the memory during a run of the Dijkstra algorithm. A perfect wear levelling approach will evenly distribute all Bit-flip peaks to the other memory regions, resulting in a flat curve.
However measuring Bit-flips for a programs execution is not a trivial task and requires the programs full memory trace. This can be realised using the Gem5 emulator and the NVMain memory simulator. Our approach also uses the Unikraft micro kernel to deploy individual tasks inside Gem5.
Not all approaches for wear levelling work on all memory regions. For example it is not trivial for a software approach to wear level the stack region without breaking a programs function calling conventions. Therefore it is necessary to target specific memory regions with the benchmark suit. In Figure 1 the operations in the BSS data region could be easily moved to heap but still using the same benchmark. This way a heap management approach could be evaluated, which could not make use of the normal Dijkstra benchmark just using on Data and BSS. Also desirable are benchmarks generating synthetic load to evaluate extreme use cases.

In this thesis, students should make themselves comfortable with our current setup to simulate Bit-flips with Gem5 and Unikraft. A desirable outcome is a Benchmark suit selectively targeting all memory regions (Data, BSS, Heap and Stack). The Benchmarks should be composed of synthetically generated load on the memory as well as real world examples like the Dijkstra algorithm shown in Figure 1.

Other suggestions and related topics are also welcome. Please do not hesitate to make an appointment.

Required Skills:

- Knowledge of computer architecture
- Knowledge of Operating Systems
- Basic knowledge of C/C++

Acquired Skills after the thesis:

- Deep Knowledge Wear levelling
- Understanding of memory distribution/regions inside a program

Literature:

- Software-Based Memory Analysis Environments for In-Memory Wear-Leveling
- Emerging NVM: A Survey on Architectural Integration and Research Challenges