CMP SCI 635: Modern Computer Architecture

NVM Duet: Unified Working Memory and Persistent Store Architecture

Name: Kunjal Panchal Date: 5th Nov, 2019 Student ID: 32126469

Paper: 16 – Secondary Storage [Liu14]

Strengths:

1. They came up with the concept of prioritizing write requests on dependency chains. A barrier delimits multiple writes into two epochs and creates dependency between the two epochs, i.e., the following epoch cannot begin execution until the preceding epoch completes. Therefore, by scheduling writes to persistent store first in the presence of barriers, they can make subsequent persistent writes available to the memory scheduler earlier, thereby exposing additional opportunities for exploiting bank-level parallelism. That improved load latency by 24% on average and increases IPC 1.28x on average.

2. Smart Refresh avoids unnecessary refreshes. If the resistance of a cell remains in the target band, the cell can retain data until the next refreshing interval because the rate of resistance drift (i.e., resistance changes in logarithm per unit of time) is time decreasing. Therefore, the refresh of this type of cell can be deferred. And we only need to worry about drifts for stage-2 and stage-3, out of total 4 stages.

Weaknesses:

1. There is no mention of what happens where all the use cases are same for which the barrier is set? Does this just describe a baseline case? Do we get any autonomy is stalling some entries of write queues to get a balanced mixture of working memory and persistent storage?

2. Upon power failure, *allocmap* will put all data in persistent store, even if it's not needed. That might increase the initial load latency drastically on the next boot. Does having a copy of use case bits somewhere fixed doesn't make sense to reduce this overhead?

Questions/Assertions:

1. Why “persistent space” just cannot be amalgamated with traditional static space and differentiated by some “p/s” kind of bit which says what uses which kind of memory?

2. Am I right in assuming that the IPC metric used in evaluation only has to deal with load and store operations?

3. How many bits at max can be stored in MLC (read about IBM making 3-bit MLC PCM)? And what will be reason of not adding more? (Maybe more density – more heat?)

4. How to deal with PCM being more temperature sensitive than MRAM? And PCM uses chalcogenide glasses, which need to be isolated from silicon to avoid contamination, do these two factors increase fabrication cost exponentially?

5. [DOI: 10.1038/s41586-018-0180-5] Research shoes PCMs can outperform GPUs in neural network training by achieving 100% test accuracy. The advantage being, we don't need to make trips back and forth to memory since the operations take place in memory, and many, many operations can be done in parallel. Those differences have natural parallels with the behaviour of a population of neurons, which makes phase-change memory a (potentially) good fit for neural networks.