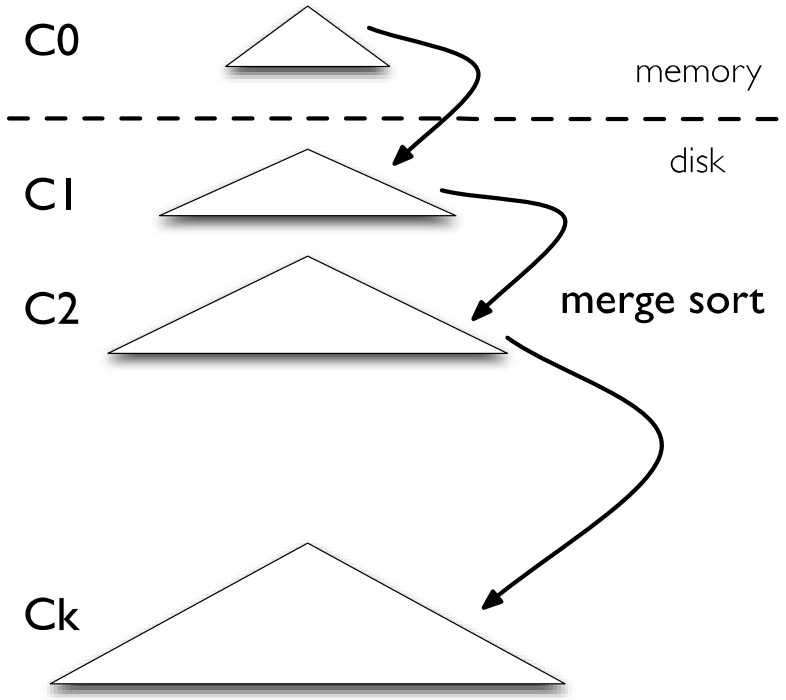
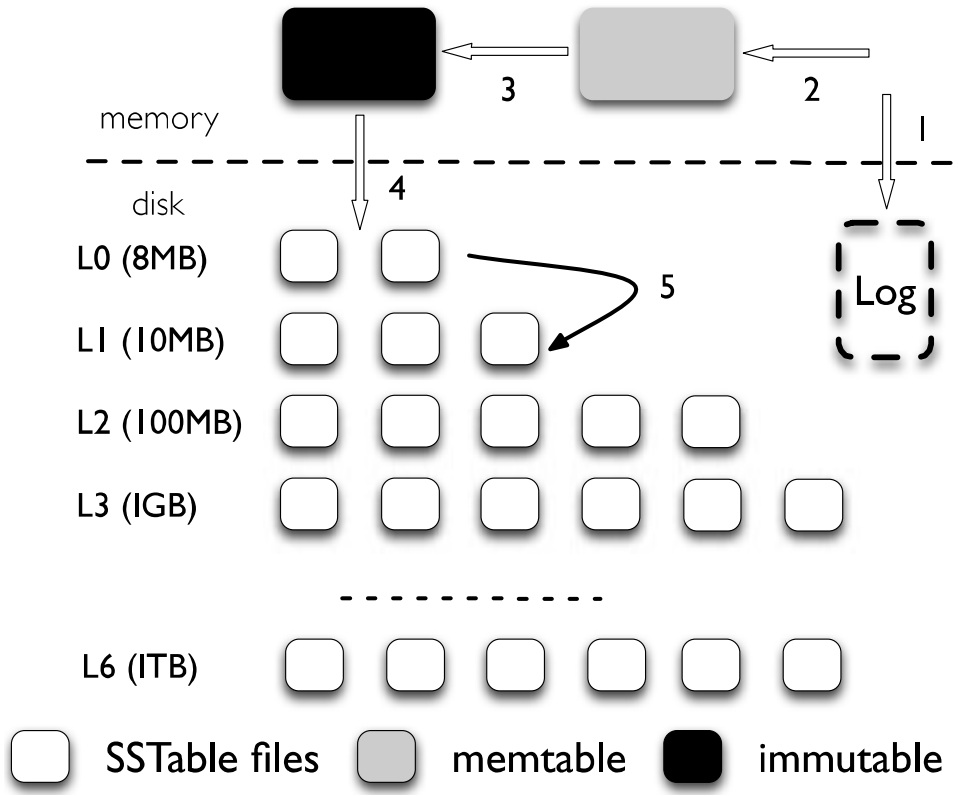
NV-LSM: Log-Structured Merging on Hybrid Volatile/Non-Volatile Memory Systems

Baoquan Zhang, Fenggang Wu, Zhichao Cao and David H.C. Du
Center for Research in Intelligent Storage
University of Minnesota – Twin Cities
Feb. 14th 2018

Why Log-Structured Merge Tree (LSM-Tree) on NVM?



LSM-Tree[1]

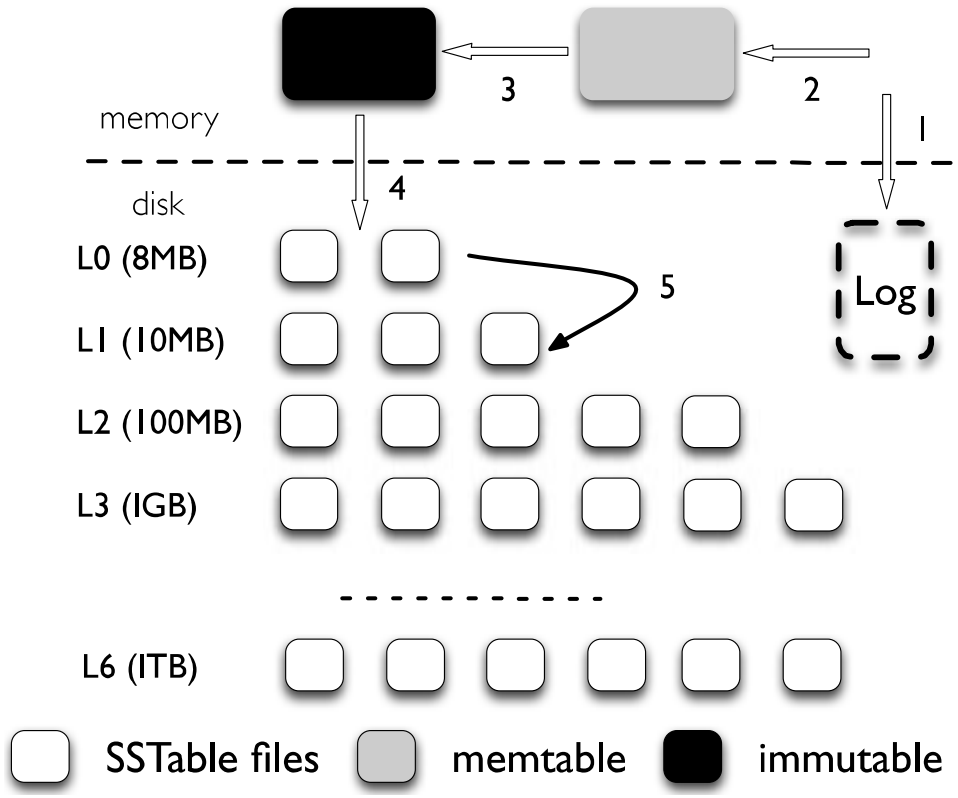


Google LevelDB[1]

[1] WiscKey: Separating Keys from Values in SSD-conscious Storage

Why Log-Structured Merge Tree (LSM-Tree) on NVM?

- Advantages
  - Good insert performance

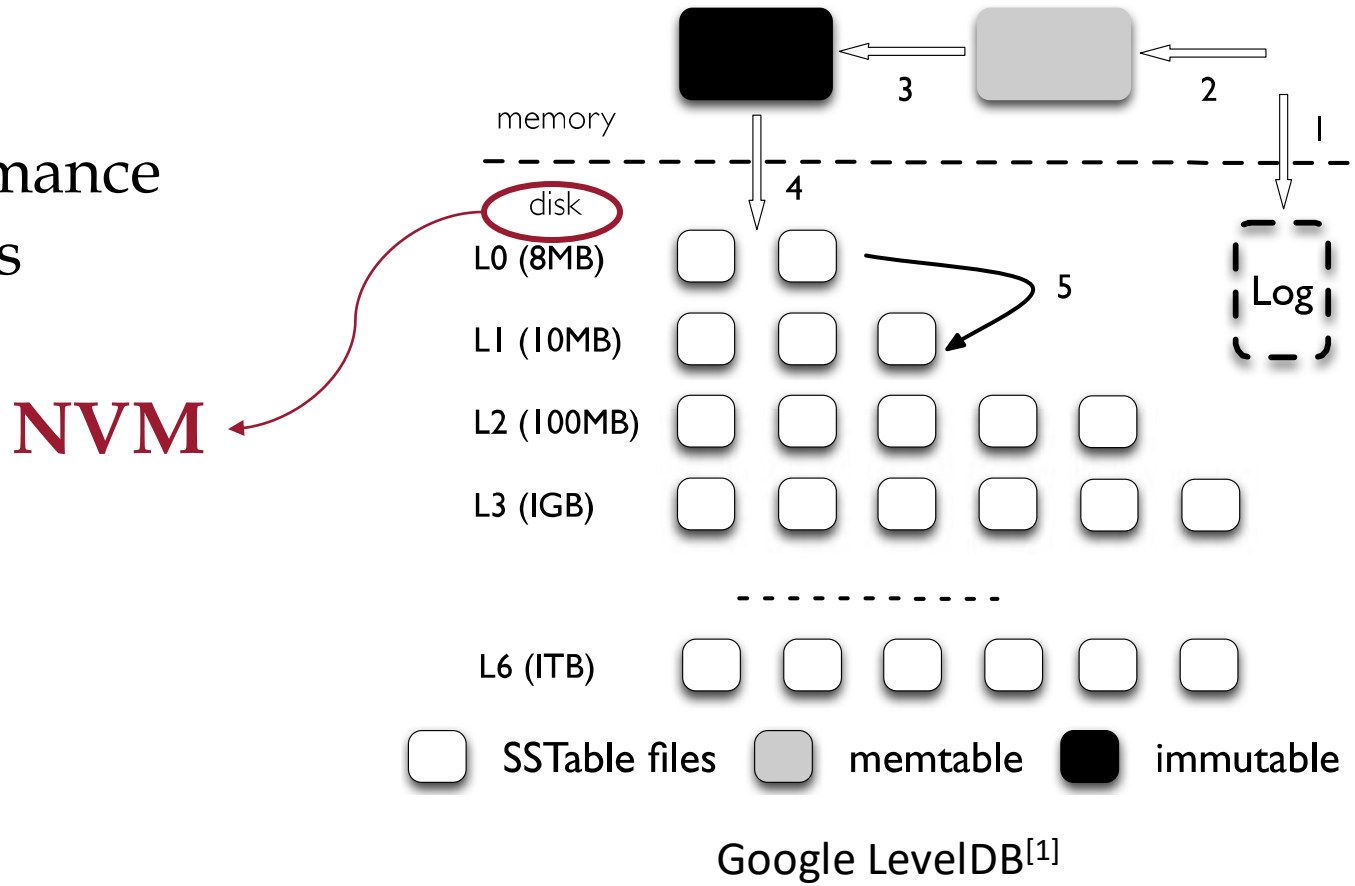


Google LevelDB[1]

[1] WiscKey: Separating Keys from Values in SSD-conscious Storage

Why Log-Structured Merge Tree (LSM-Tree) on NVM?

- Advantages
  - Good insert performance
  - No in-place updates



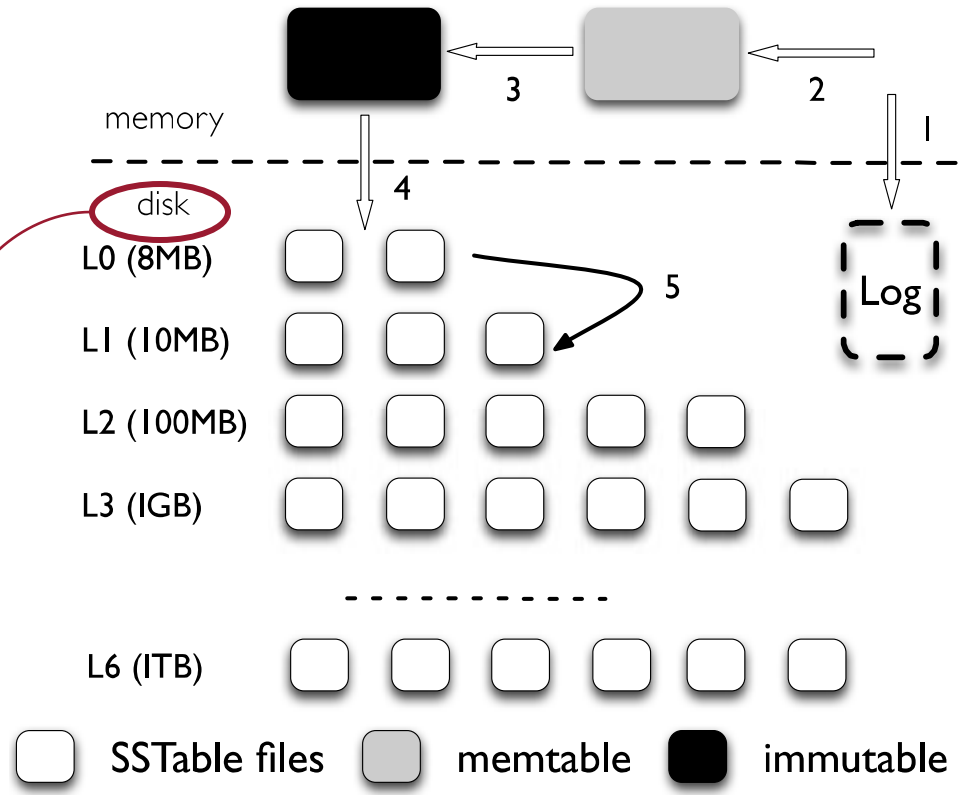
Google LevelDB[1]

[1] WiscKey: Separating Keys from Values in SSD-conscious Storage

Why Log-Structured Merge Tree (LSM-Tree) on NVM?

- Advantages
  - Good insert performance
  - No in-place updates
- Problems
  - Compaction overheads
    - Data re-reads/re-writes

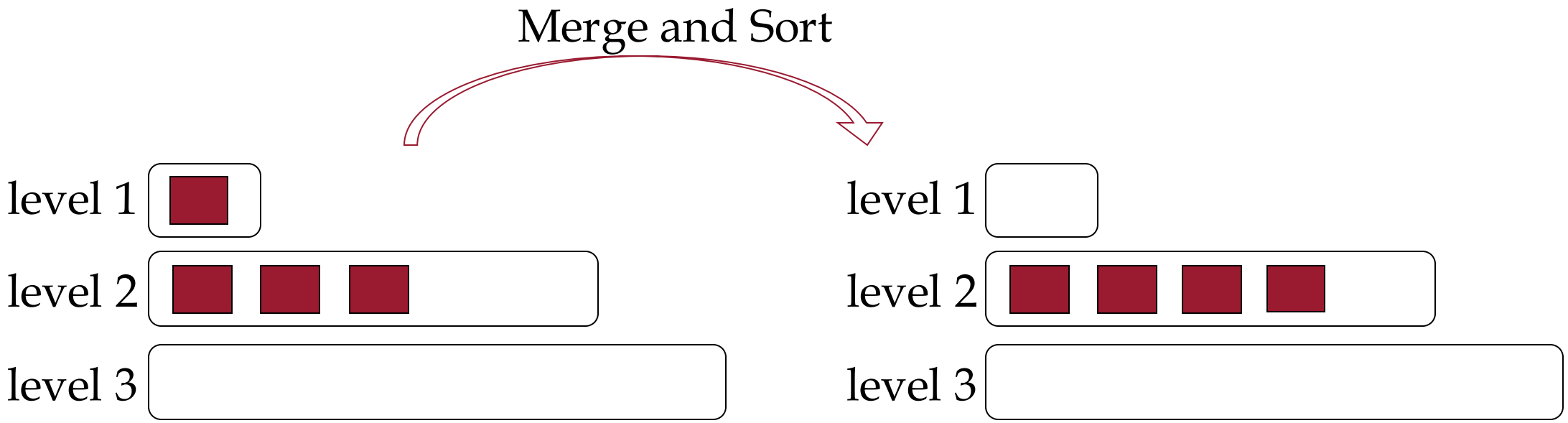
NVM



Google LevelDB[1]

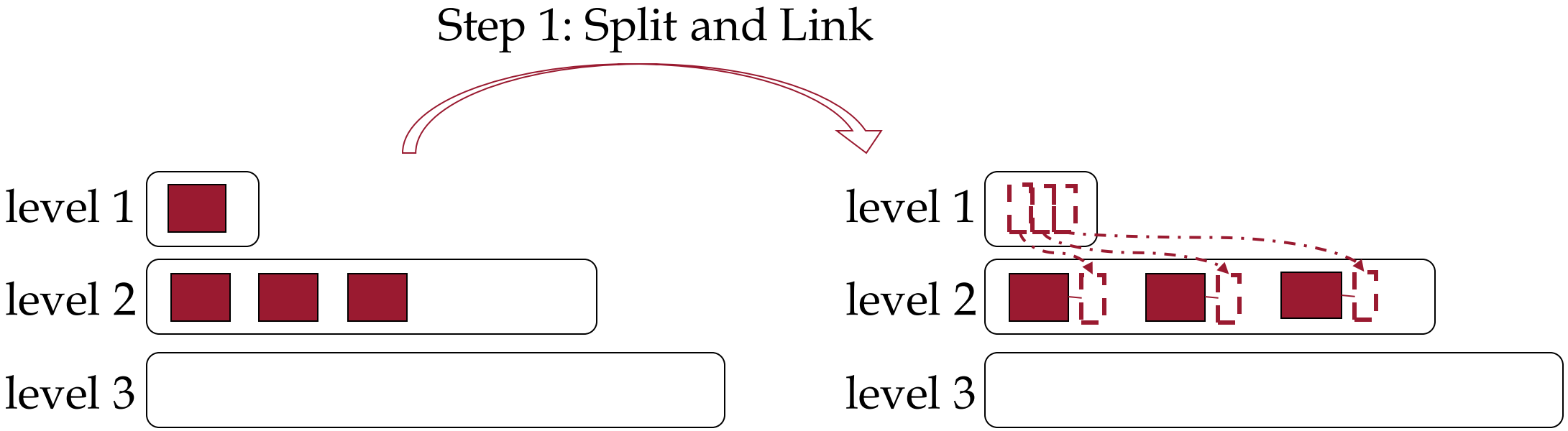
[1] WiscKey: Separating Keys from Values in SSD-conscious Storage

Compaction Overheads Introduced by Data Re-Reads/Re-Writes



Data will be re-read & re-written ***Multiple Times*** before moved to next level.

Proposed Approach: Lazy-Compaction



Step 2: Merge with two conditions

(1) Pre-Defined Maximal Segments

(2) Compact to Next Level

Thanks for listening!
Q & A