Employing NVM for Non-blocking Segment Cleaning in F2FS

Changgyu Lee, **Hyunki Byun**, Sungyong Park, Youngjae Kim

Sogang University, Seoul, Korea

FAST'18 WIP

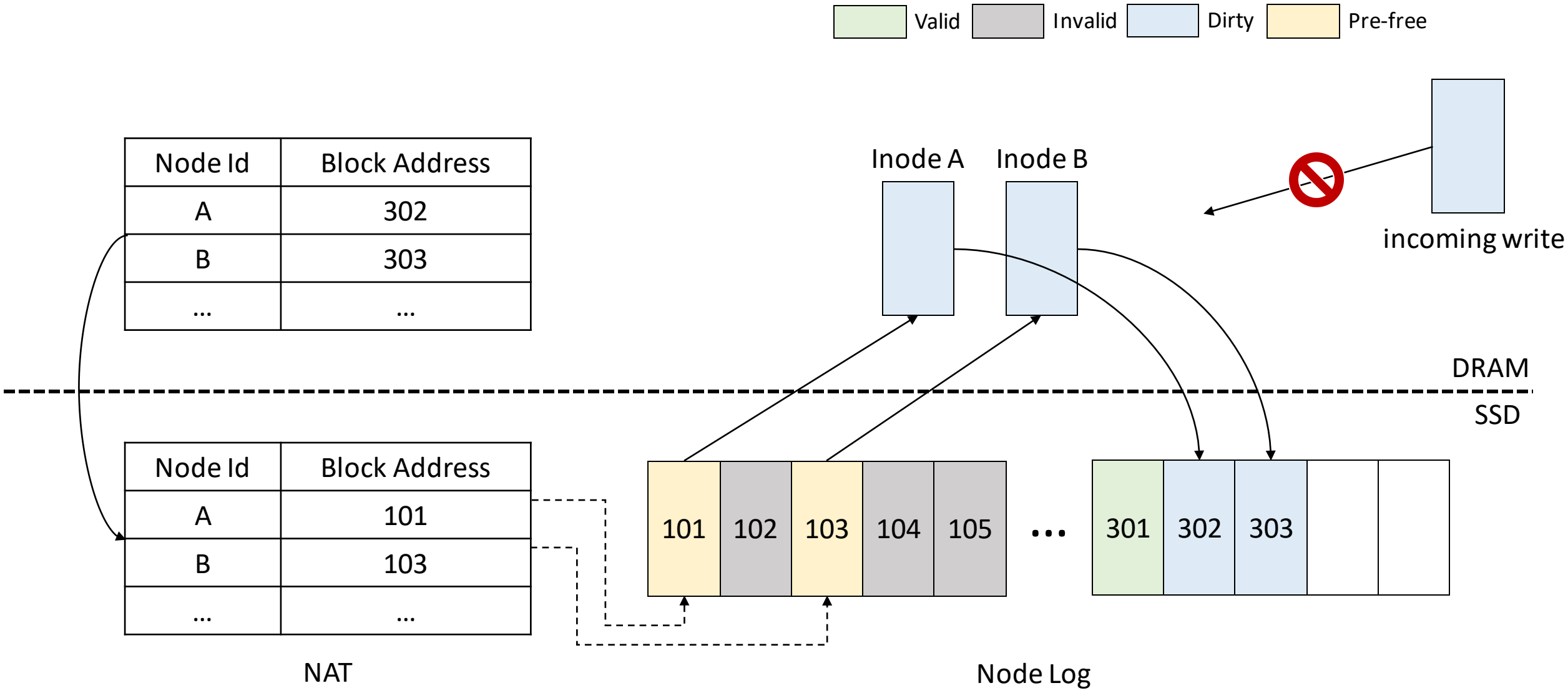


Background

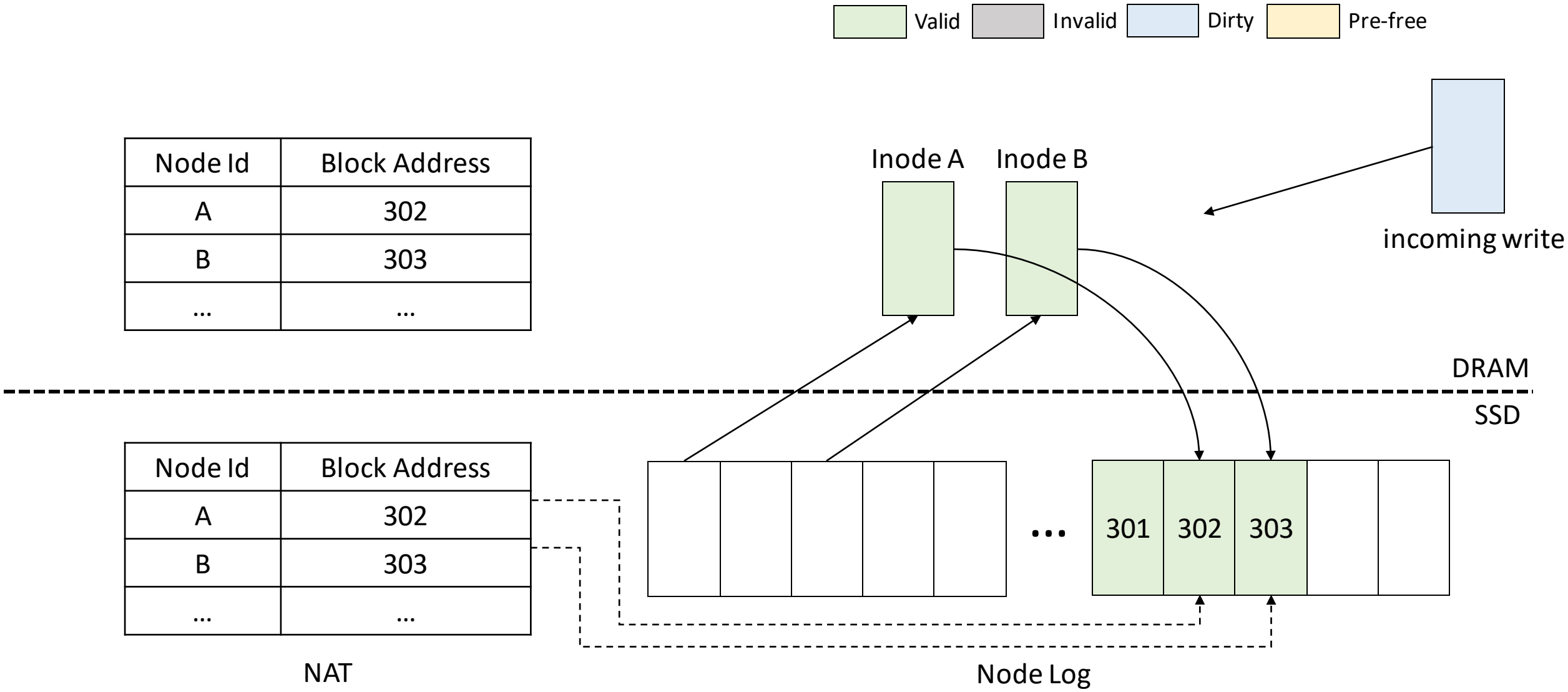
- F2FS employs two types of logs to benefit with Flash parallelism and garbage collection
  - Data log for directory entry and user data
  - Node log for inode and indirect node
- Node Address Table (NAT) translates node id to block address
- NAT entry's block address is updated when corresponding data is flushed to the storage device
- In order to re-claim free log space, F2FS initiates segment cleaning which in turns trigger checkpointing

However, checkpointing blocks incoming write operations, which degrades the file system performance

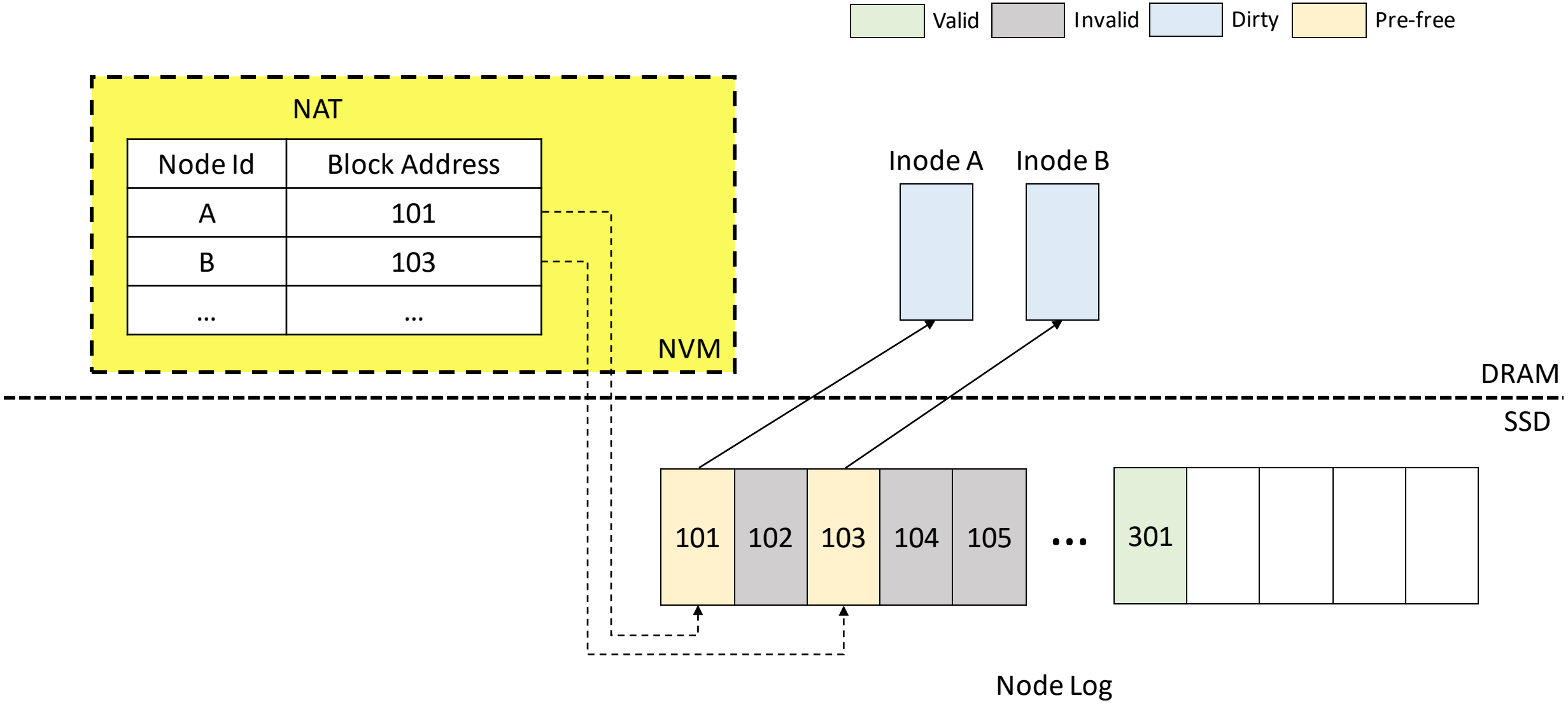
Segment Cleaning in F2FS



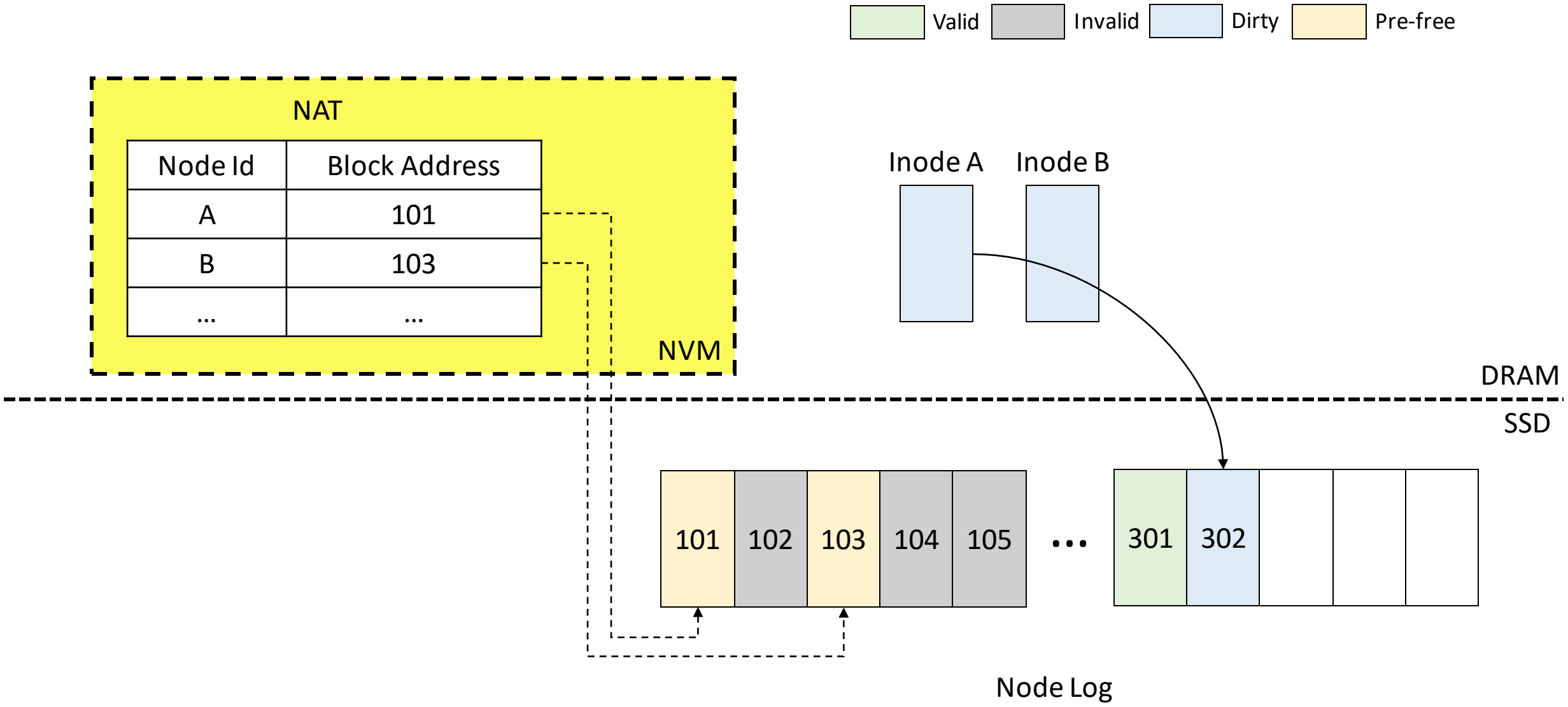
Segment Cleaning in F2FS



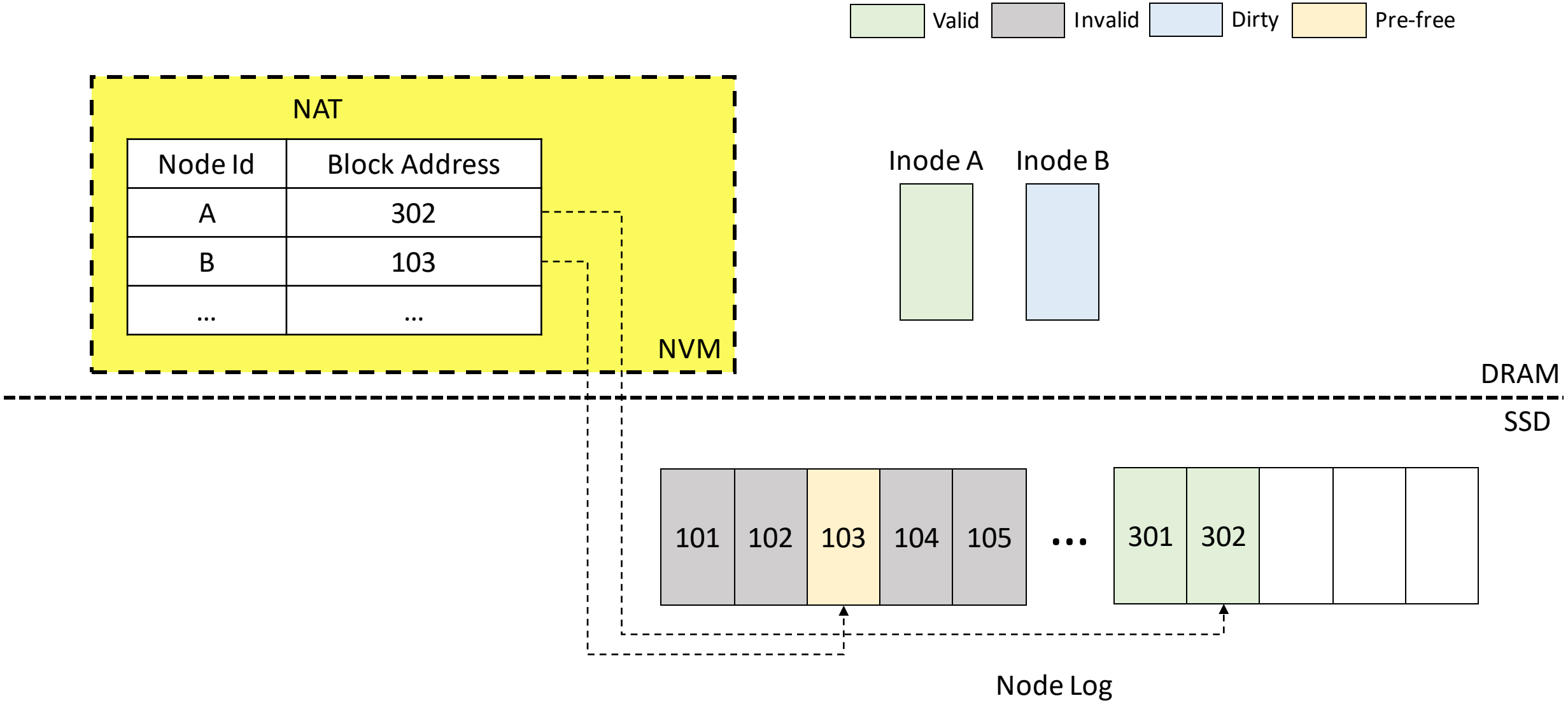
Segment Cleaning in F2FS



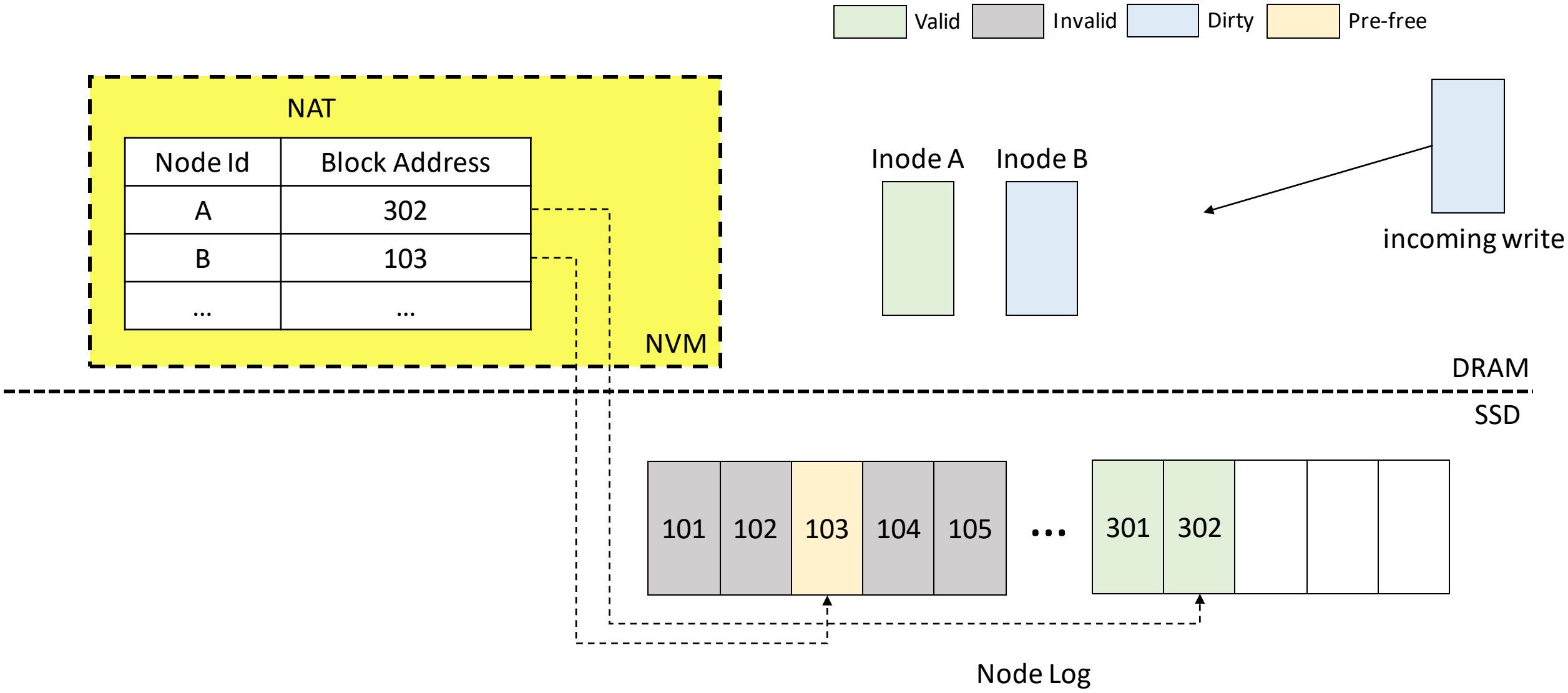
Segment Cleaning in F2FS



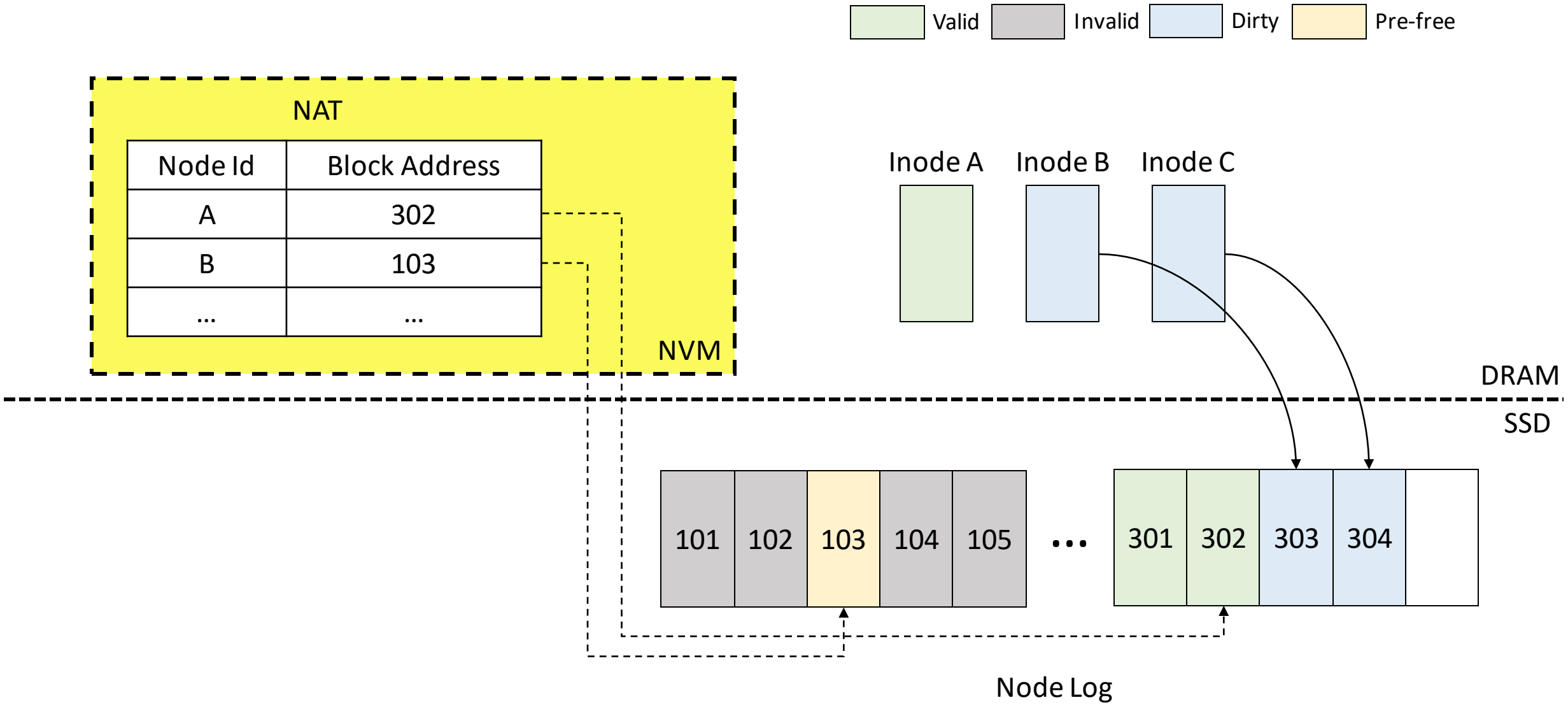
Segment Cleaning in F2FS



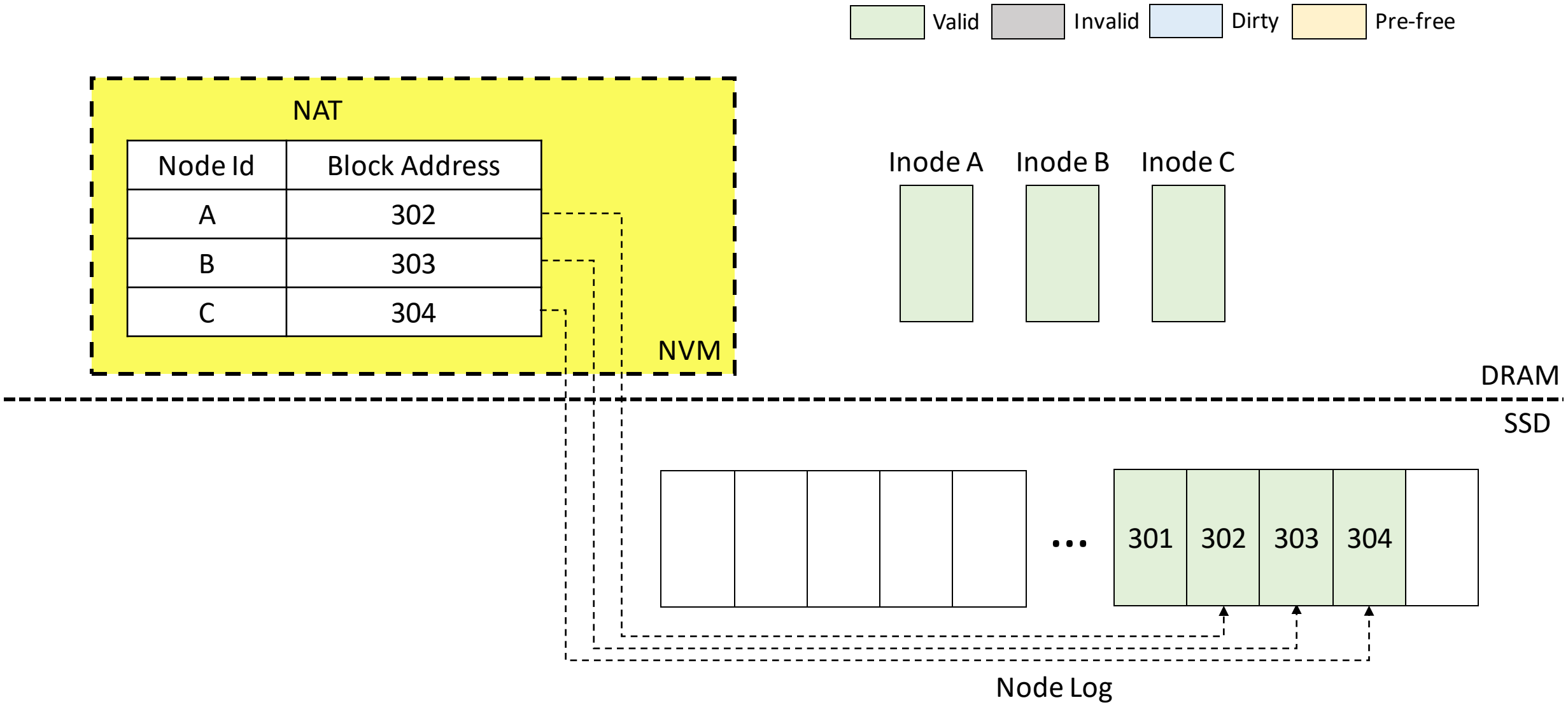
Segment Cleaning in F2FS



Segment Cleaning in F2FS



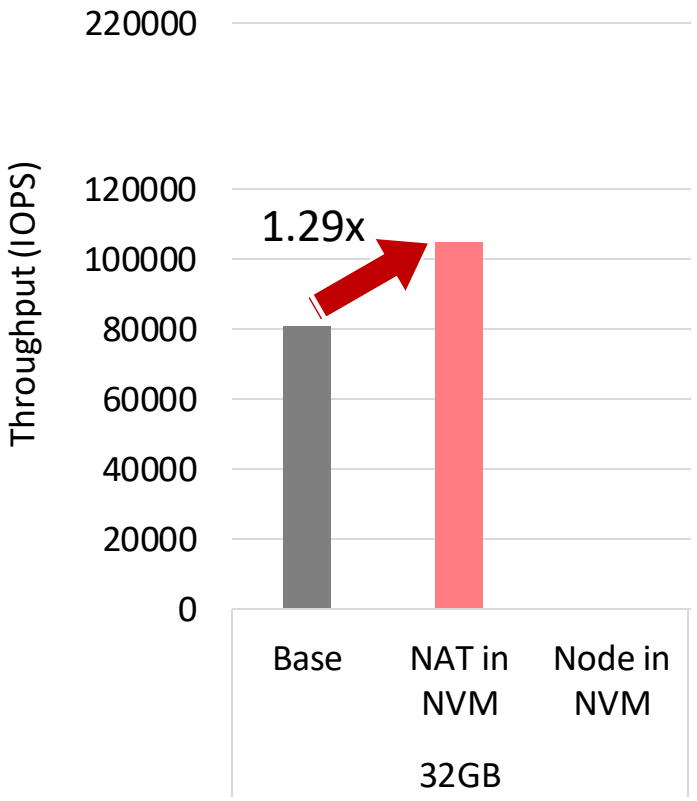
Segment Cleaning in F2FS



Experimental Setup

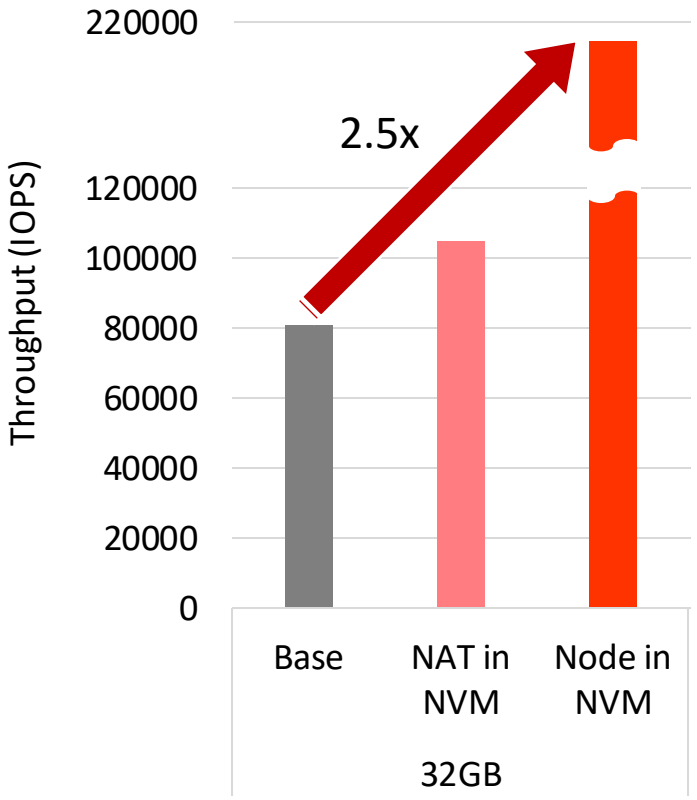
- Host Machine
  - 10 CPU cores, 32GB DRAM, 400GB NVMe SSD and Linux kernel v4.14.11
- Non-Volatile Memory (NVM)
  - Emulated a portion of main memory as NVM
- Workloads
  - Modified the fxmark for workloads that cause frequent segment cleaning
  - Each core issues writes, each of which is 4KB overwrite followed by fsync
- We measured throughput (IOPs) for 60 sec after the SSD was filled to 90%

Preliminary Result



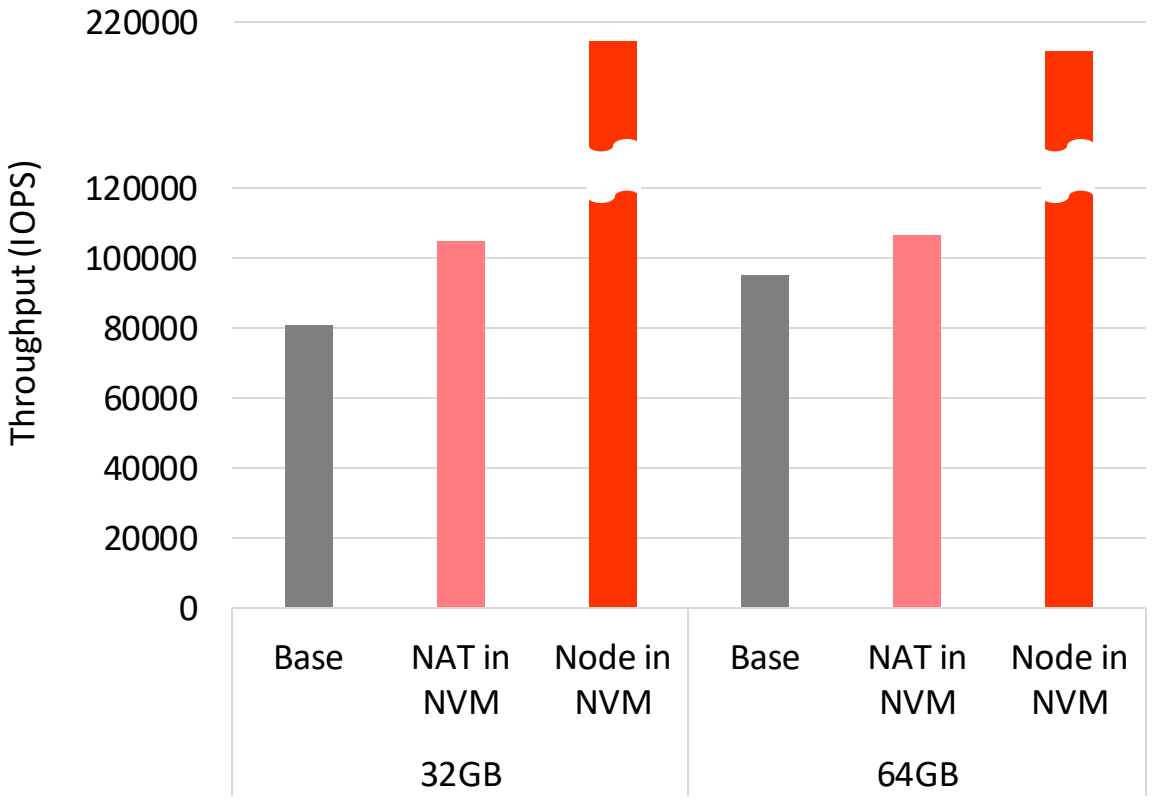
- When we migrate NAT to NVM, throughput increased from 80953 to 104743, about 1.29 times

Preliminary Result



- When we migrate NAT to NVM, throughput increased from 80953 to 104743, about 1.29 times
- When we migrate NAT and Node log to NVM, throughput increased from 80953 to 209835 about 2.5 times

Preliminary Result



- When we migrate NAT to NVM, throughput increased from 80953 to 104743, about 1.29 times
- When we migrate NAT and Node log to NVM, throughput increased from 80953 to 209835 about 2.5 times
- We have similar observations with 64GB partition

Thank You