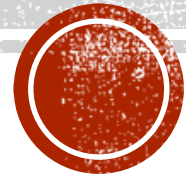


PRINCIPLES OF DATA ORGANISATION

BFTL



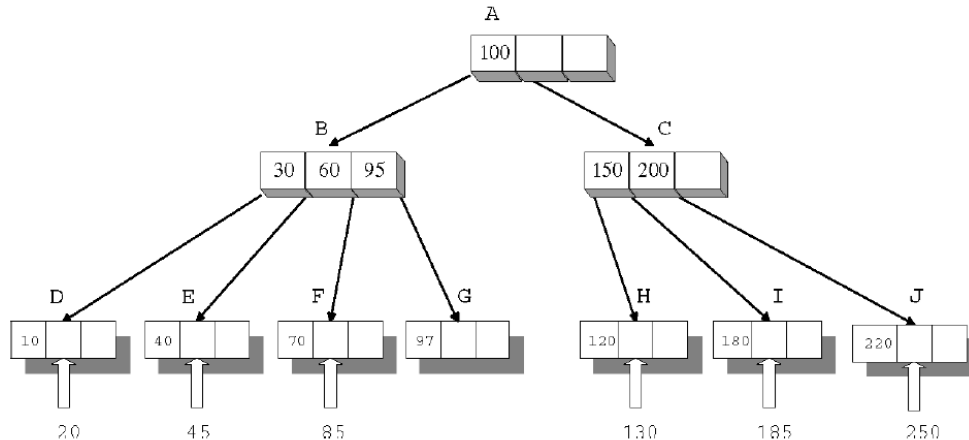
MOTIVATION

- ↳ SSD with FTL
 - ↳ Flash translation layer
- ↳ Inserting a single record causes a whole page copy
 - ↳ We want to minimize the updates



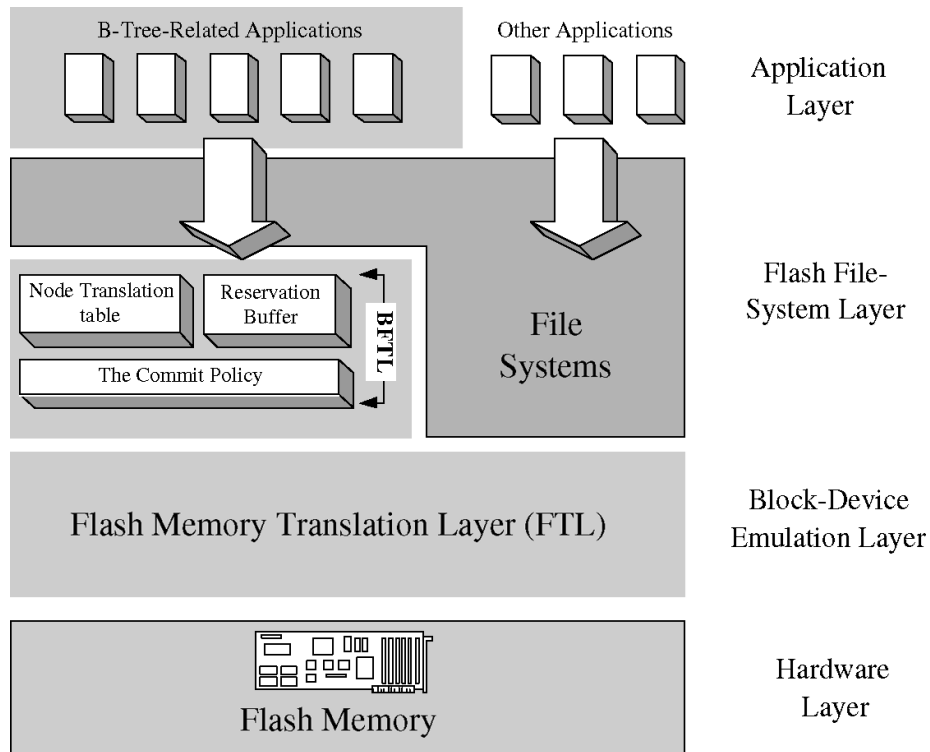
BFTL

- Wu and Kuo, 2007 : B-tree index management over flash-memory storage systems
- Motivation built over FTL inserting a single record causes whole page copy (possibly more when rebalancing is needed)
- Leads to higher free space consumption and unnecessary garbage collection
- Idea: Let us buffer modifications and submit them in bigger chunks

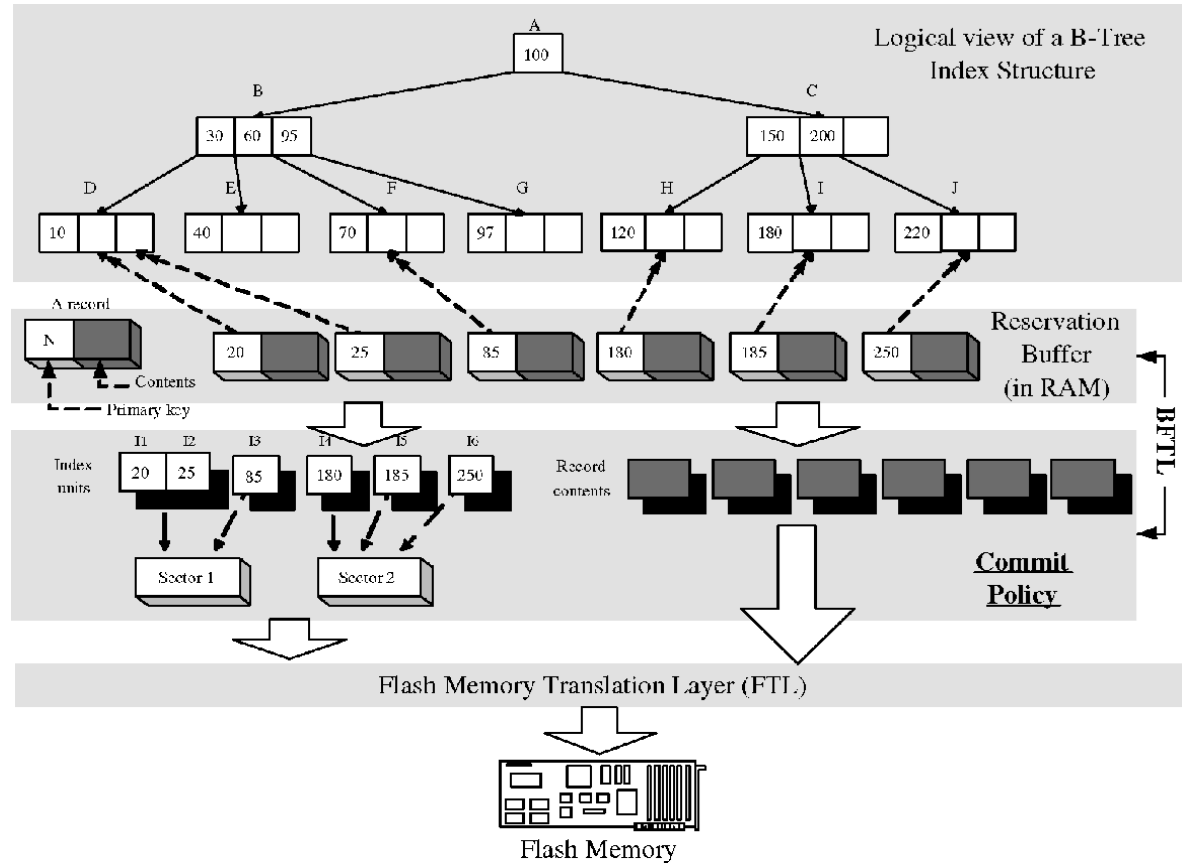


BFTL ARCHITECTURE

- ❧ B-tree index services requested by the upper-level applications are handled and translated from file systems to BFTL and then block-device requests are sent from BFTL to FTL
- ❧ BFTL was meant to be a part of the operating system



BFTL INSERT



BFTL SEARCH

- ❧ BFTL places a new layer on the flash memory file system composed of **Reservation Buffer (RB)** and **Node-Transition Table (NT)**.
- ❧ Modification operations are gathered in the RB and once a time flushed to as little segments as possible
 - ❧ According to the **Commit Policy**
- ❧ Insert into a **node** can cause the content of the node to exist in **multiple segments/pages** over flash memory.
- ❧ NT logically binds node entries being distributed through different pages (linked list of pages where the node actually resides).
 - ❧ To construct a logical node requires multiple read operations.
 - ❧ To prevent uncontrollable growth of the chains a threshold is set which, when reached, causes given list to compact.

