Removing the Costs and Retaining the Benefits of Flash-Based SSD Virtualization with FSDV

Yiying Zhang

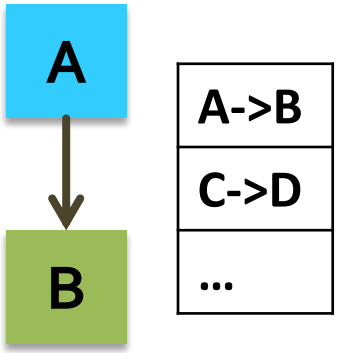
Andrea Arpaci-Dusseau, Remzi Arpaci-Dusseau

UCSDCSE
Computer Science and Engineering

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Indirection (Virtualization)

- Indirection: Reference an object with a different name
  - Page table
  - Flash Translation Layer
  - Remapping tables for HDDs, RAID, etc…

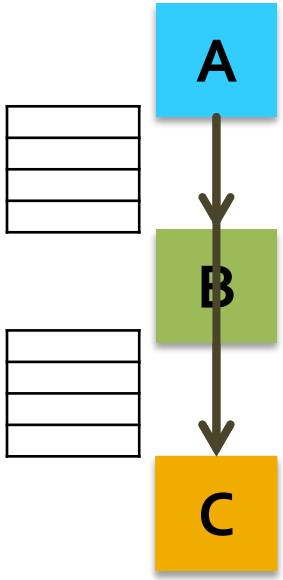


- Benefits of indirection
  - Simplicity, flexibility, modularity, uniform interface…

Indirection is good. But have we taken it too far?

Too Much Indirection

- Excess indirection
  - Redundant levels of indirection in a system
  - e.g., file system on top of SSD
- Cost of excess indirection
  - (RAM) space cost for mapping tables
  - Performance cost

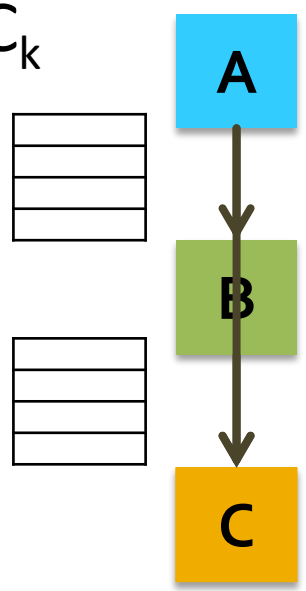


Excess indirection is redundant and costly!

How can we make systems work more efficiently?

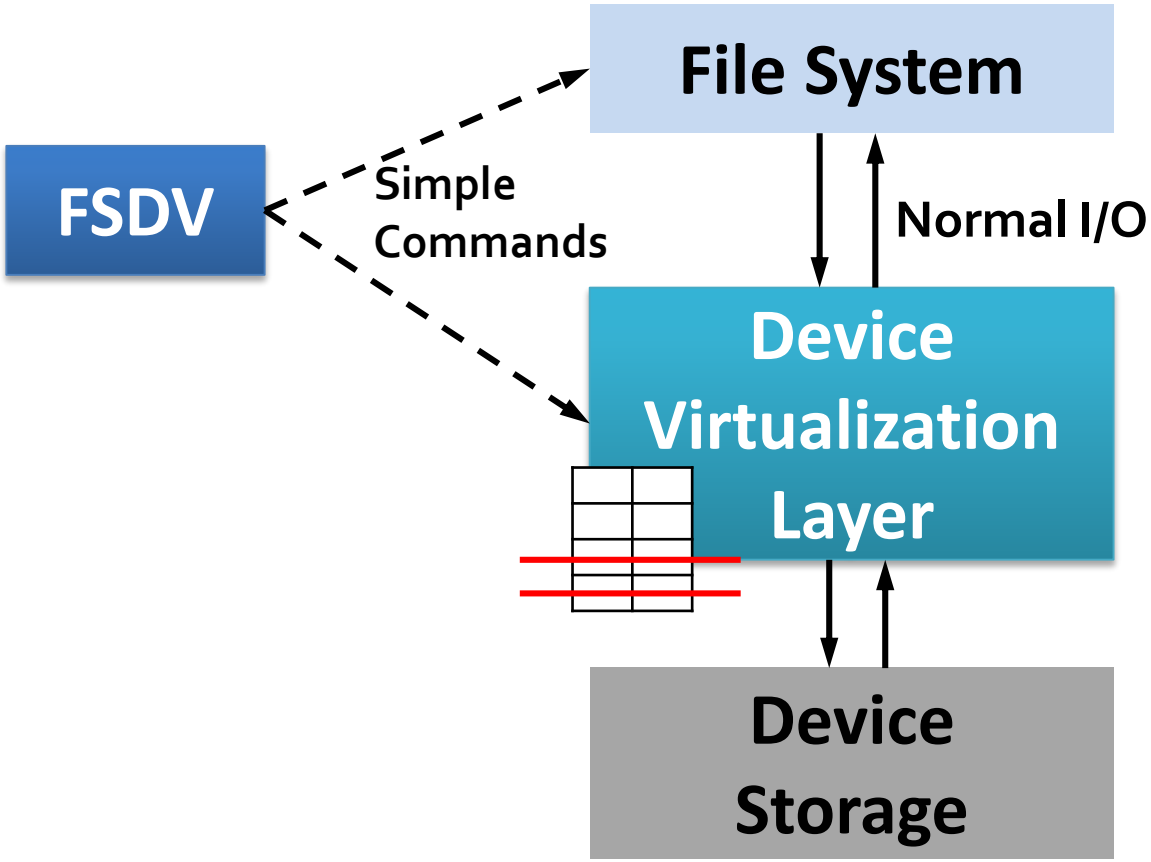
De-indirection

- Remove excess indirection
  - Collapsing redundant mappings
  - $F(A_i) = B_j$; $G(B_j) = C_k$ ➡ $H(A_i) = G(F(A_i)) = C_k$



- Example of de-indirection
  - Nameless Writes
- Problem of Nameless Writes
  - Major changes to FSs, devices, and I/O interface
  - Have to perform de-indirection for all writes

FSDV - File System De-virtualizer

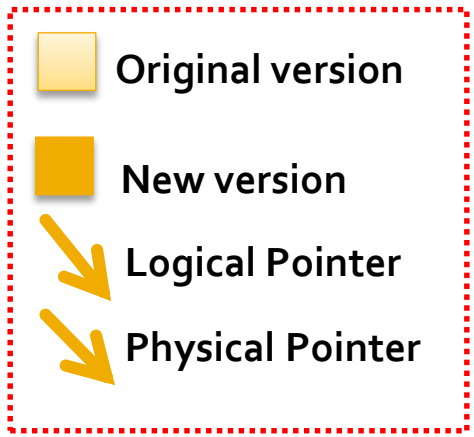


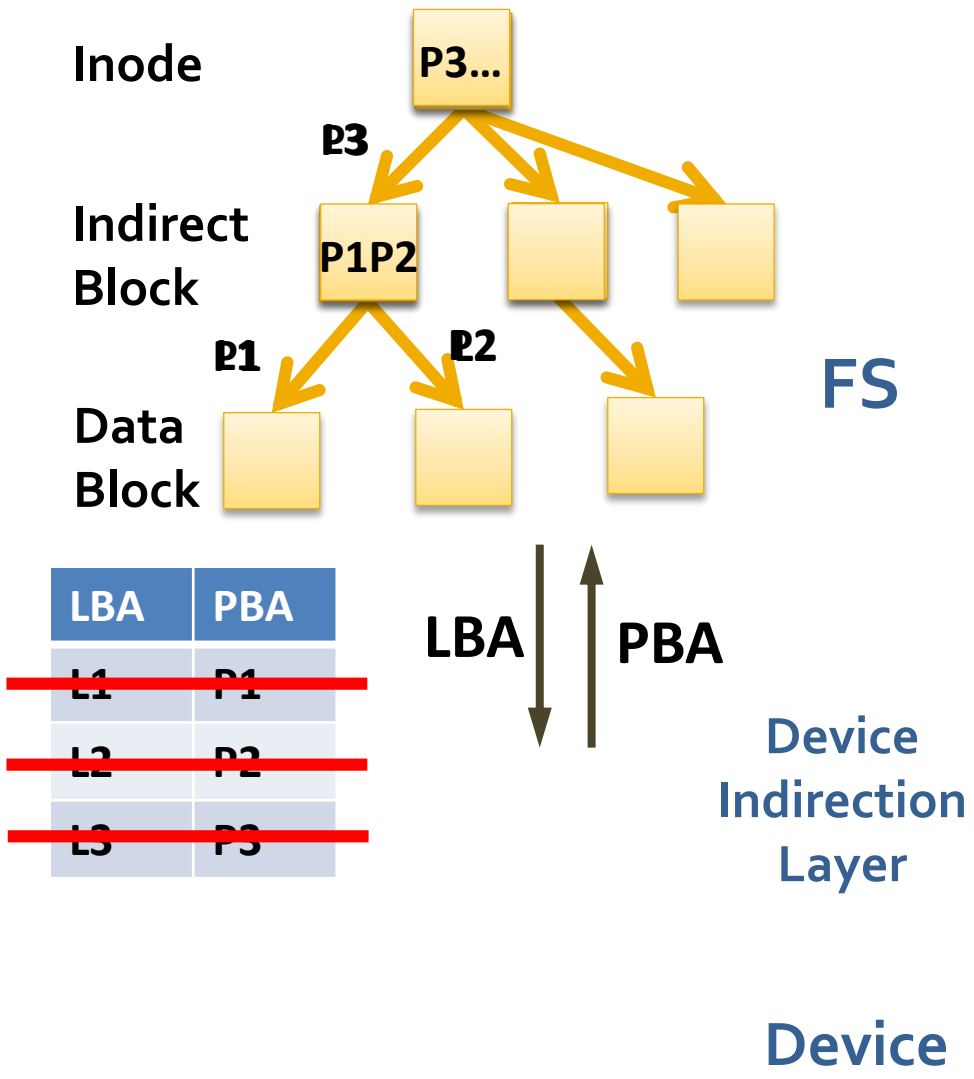
FSDV - File System De-virtualizer

- Idea: change file system pointers to physical addr
  - Walk through file system structures
  - Change pointers to device physical addresses

- Benefits
  - Dynamically remove indirection
  - Small changes to file systems and devices
  - Work with current I/O interface

FSDV Process



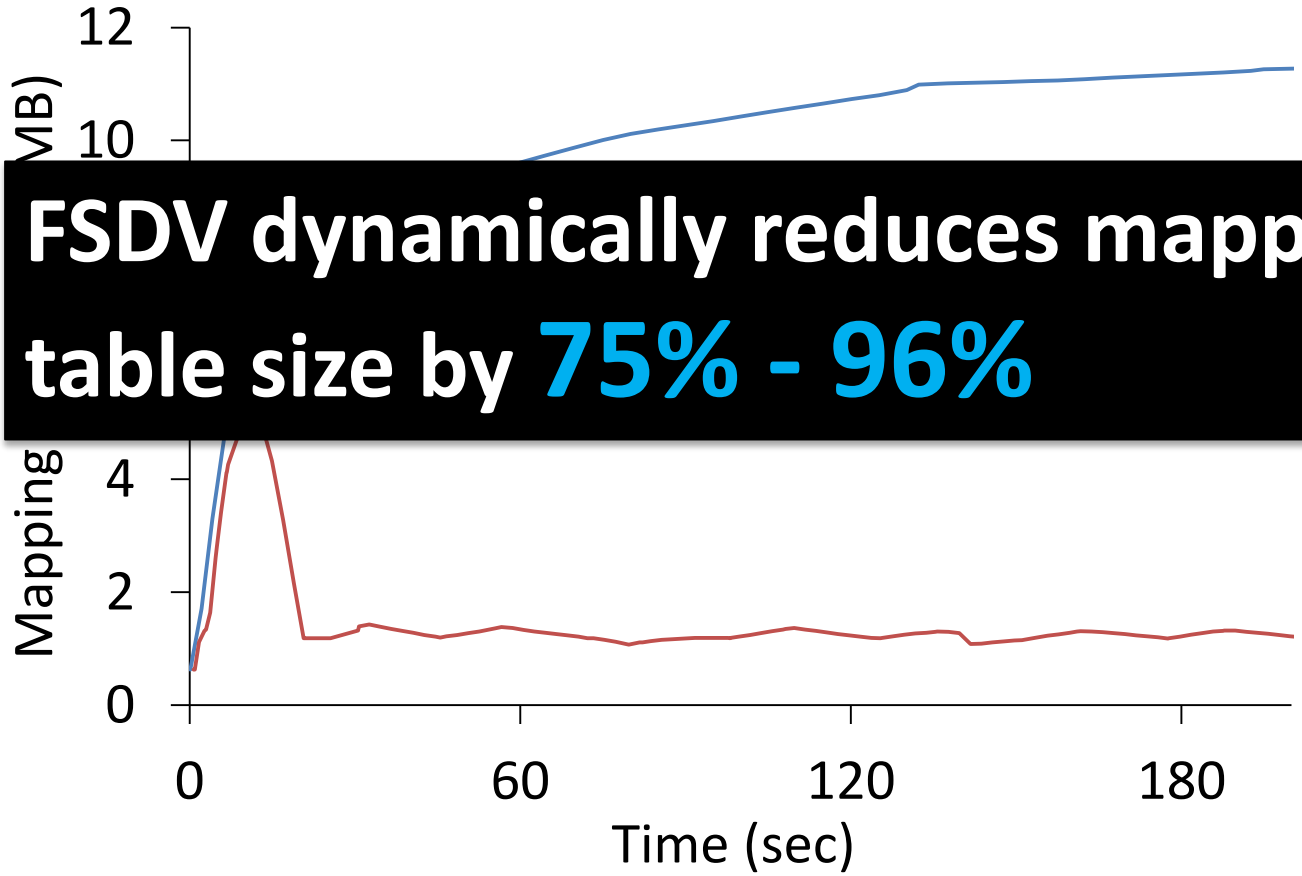


Implementation

- FSDV implemented as user-level tool
  - Implemented using *fsck* as code base
- Changes in ext3 and OS block layer
  - Total lines of code: **201**
  - Total lines of code for Nameless Writes: **4370**
- Changes in emulated SSD
  - Based on page-level FTL
  - Supports for FSDV
  - Dynamic mapping table

Mapping Table Size Reduction

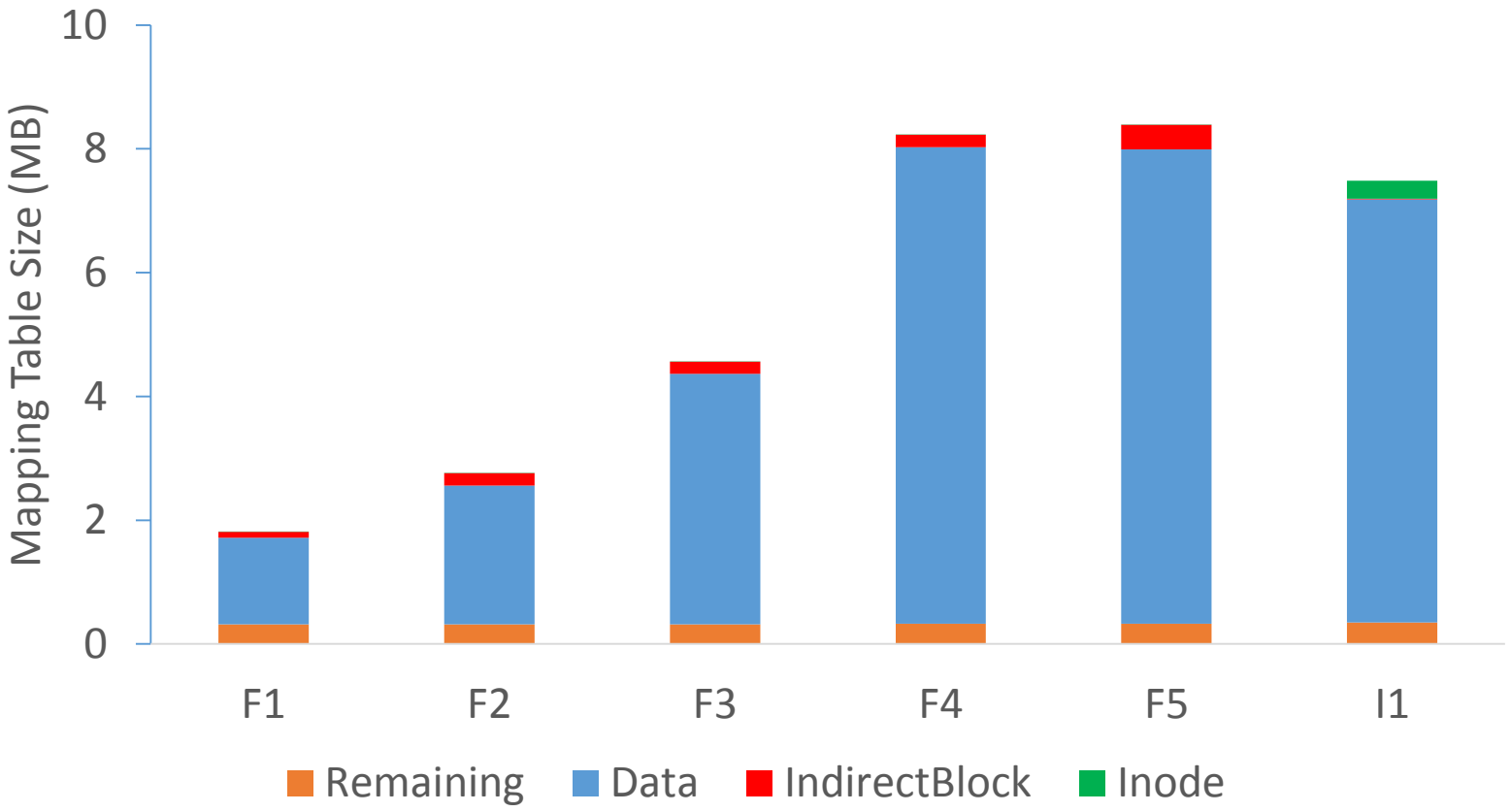
- FileServer workload
  - 2G file system, 2000 files, avg file size 1MB
  - Repeat the workload each 1 min, offline FSDV invoked in b/w



FSDV dynamically reduces mapping table size by 75% - 96%

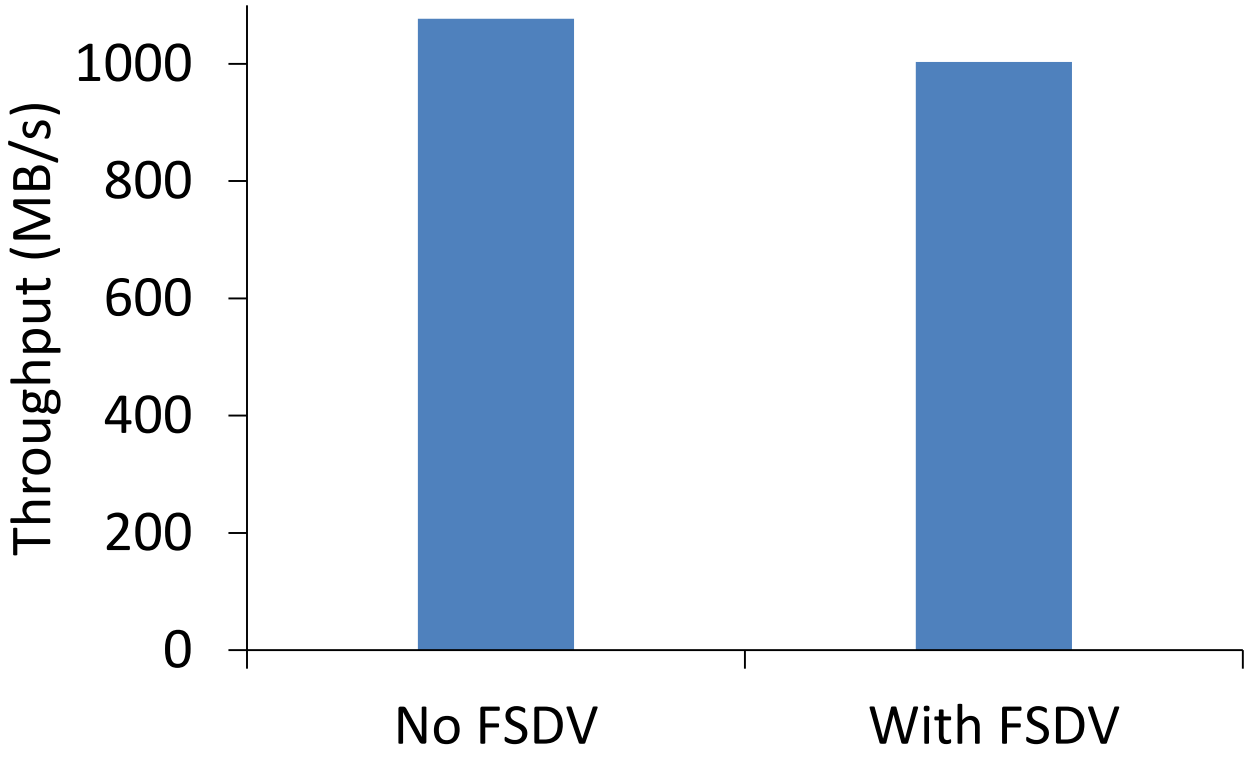
Mapping Table Size Reduction

- FileServer and Impressions workloads



FSDV Performance Overhead

- Foreground I/O throughput compare to a scheme with great performance but huge mapping table space



Summary

- Tool to de-virtualize file system pointers
- Dynamically reduce SSD mapping table size
- Small overhead to foreground I/Os
- Small file system and device changes
- Can integrate into current I/O interface

Thank you !
Questions ?

yiyingzhang@cs.ucsd.edu

UCSDCSE
Computer Science and Engineering

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON