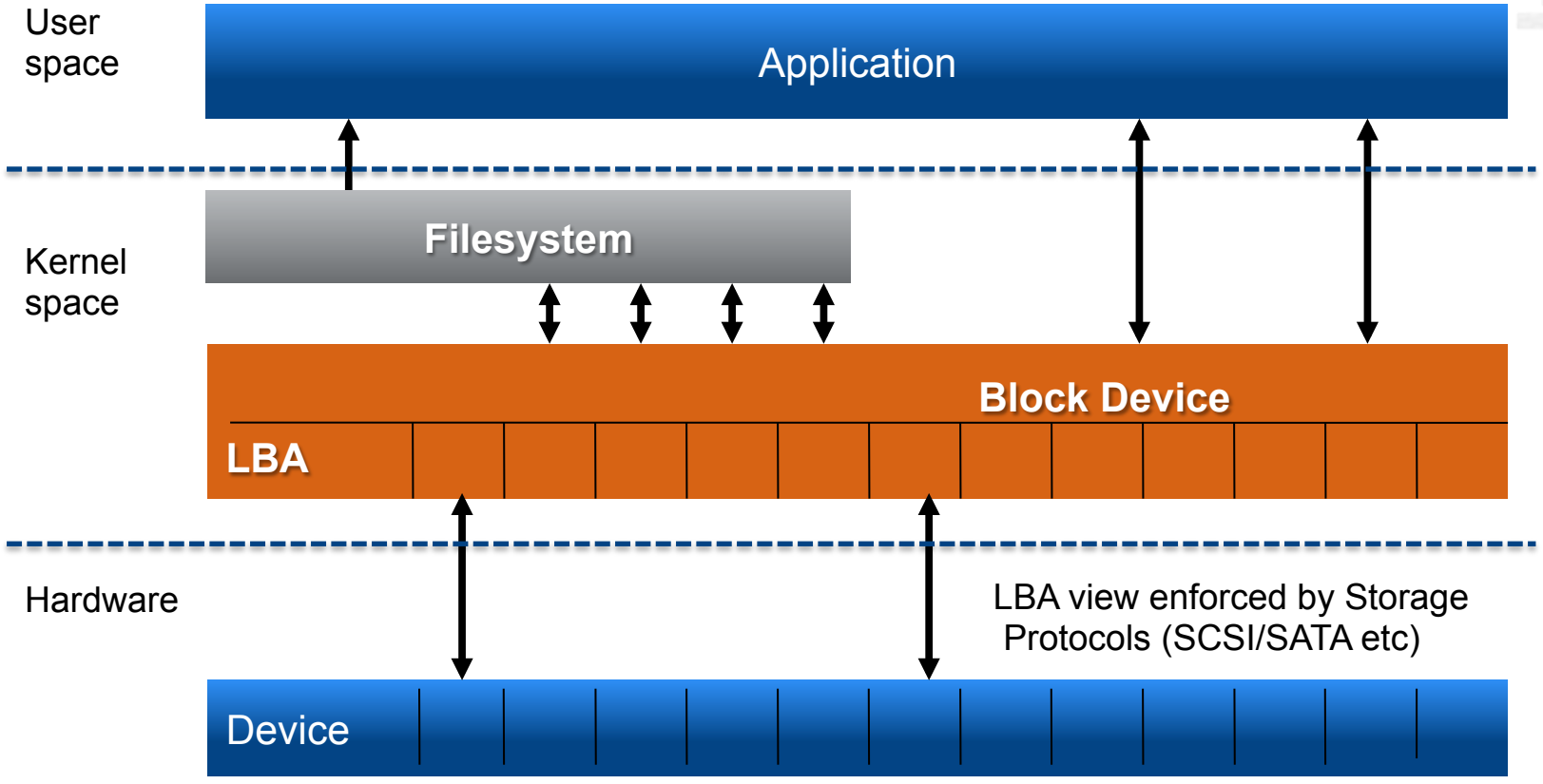
NATIVE FLASH SUPPORT FOR APPLICATIONS

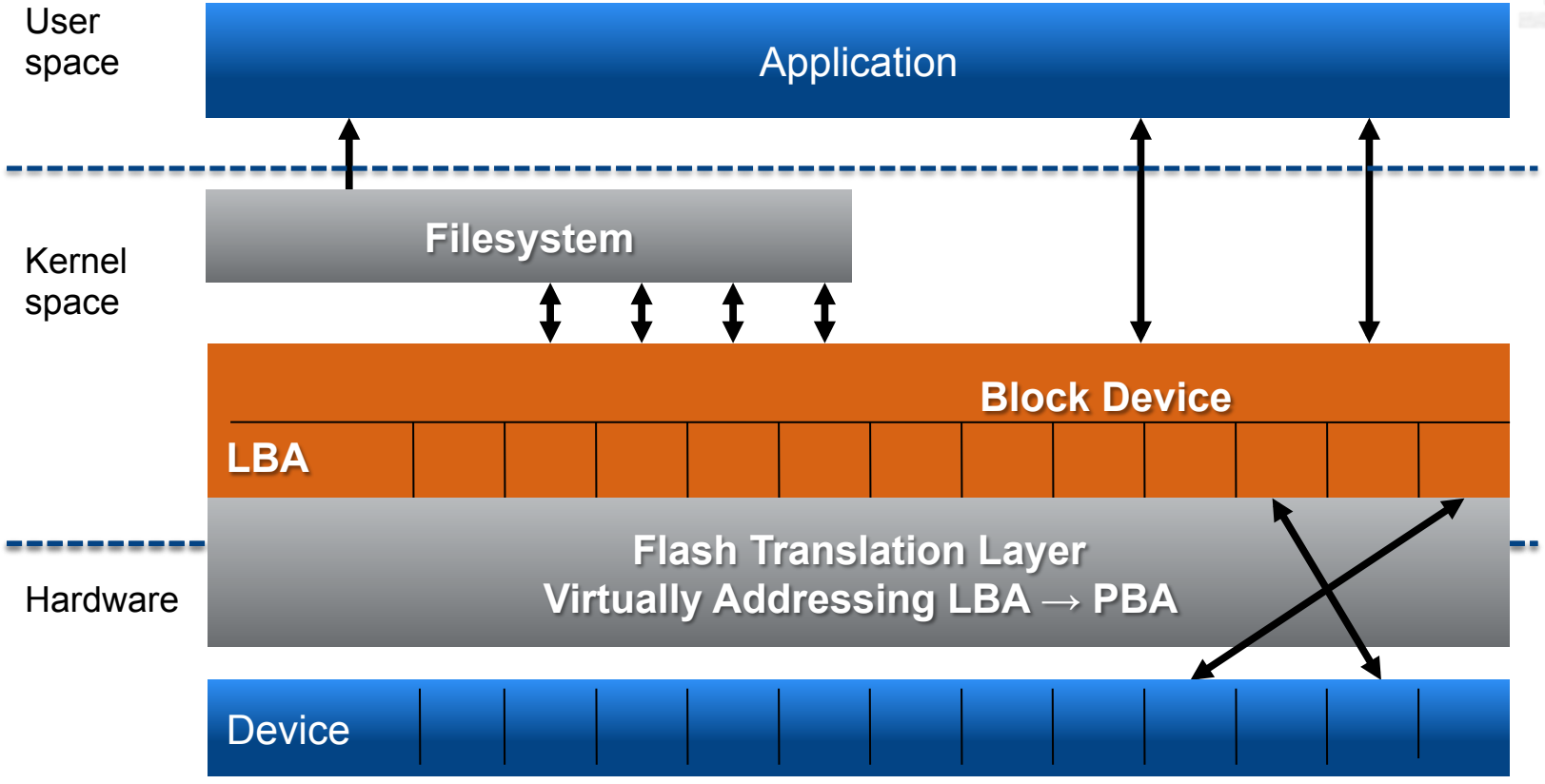
Nisha Talagala

TRADITIONAL STORAGE STACK



FLASH IN TRADITIONAL STORAGE STACKS

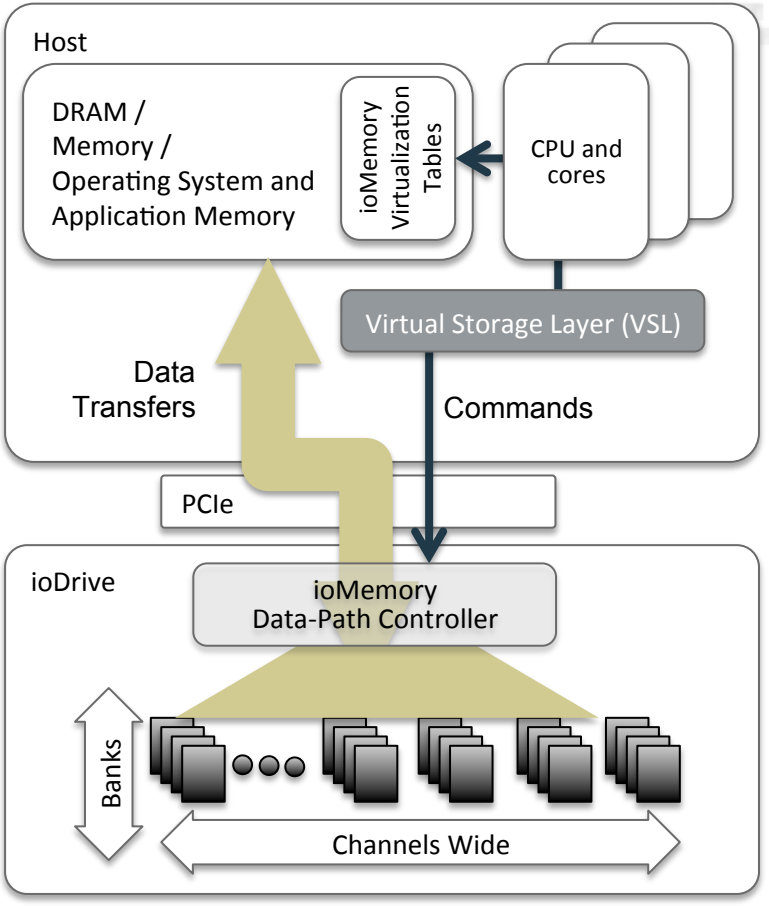
FUSION-IO



VIRTUAL STORAGE LAYER

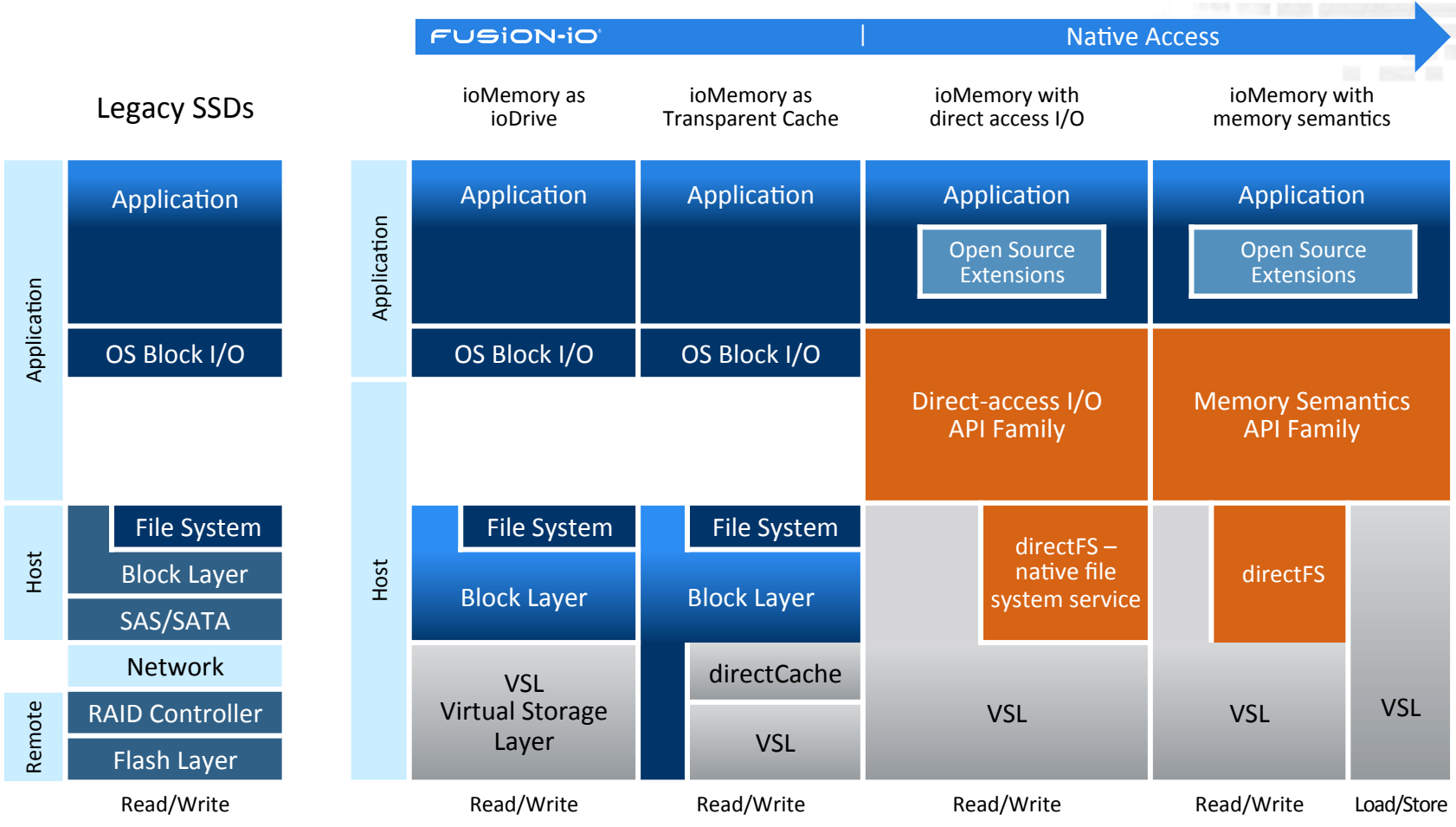
FUSION-IO

- Cut-thru architecture – avoids traditional storage protocols
- Scales with multi-core
- HW/SW functional boundary defined as optimal for flash
- Traditional block access methods for compatibility
- New access methods, functionality and primitives natively supported by ioMemory



FLASH MEMORY EVOLUTION

FUSION-IO



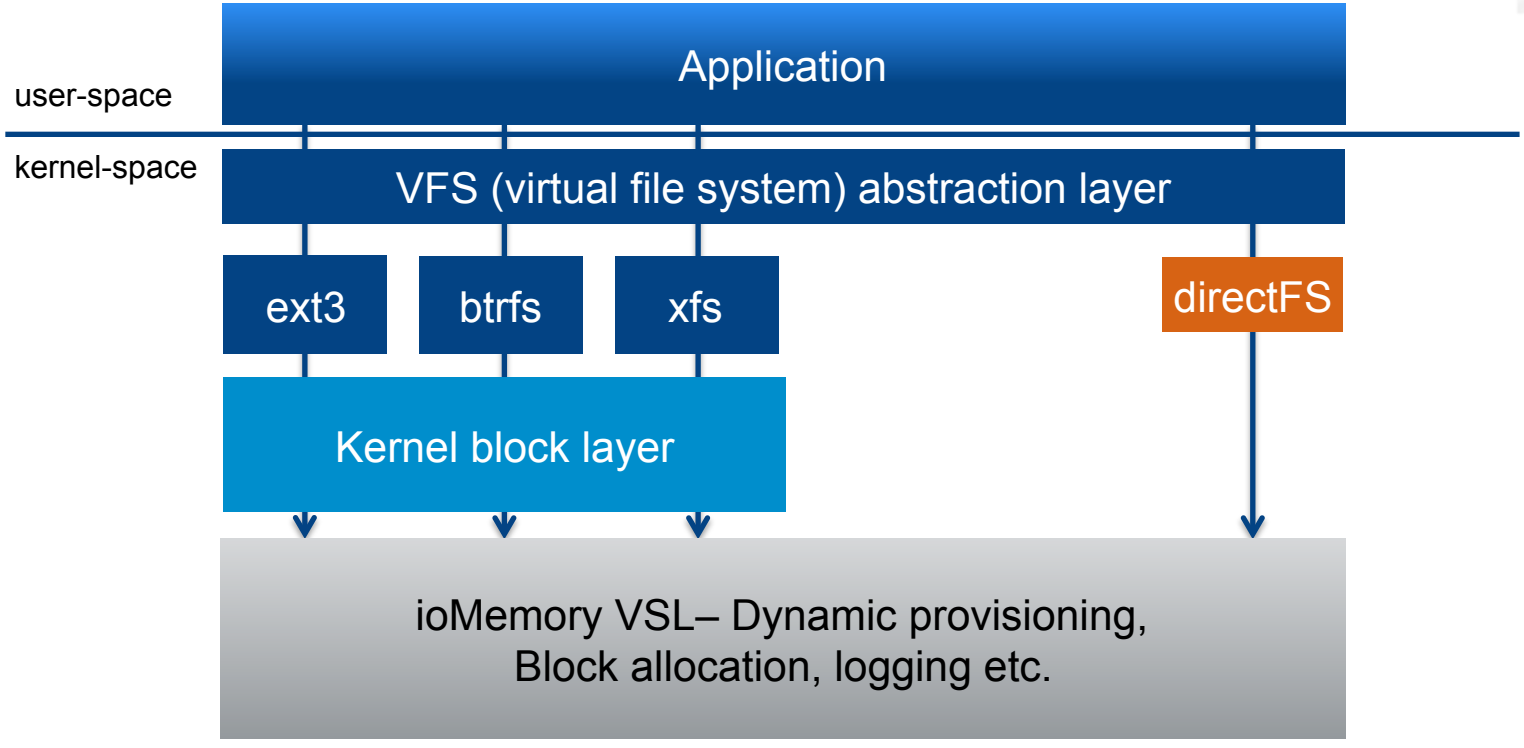
EXPLOITING NATIVE CHARACTERISTICS OF IOMEMORY

FUSION-IO

1. Native log-append writes
   incorporates **copy-on-write** basics
2. Native block mapping and allocation
   incorporate **file system** basics
3. Native large virtual address space
   incorporates **sparse semantics**
4. Native storage methods
   incorporate **key-value store** basics

DIRECTFS – NATIVE FILE NAMESPACE FOR FLASH

FUSION-IO



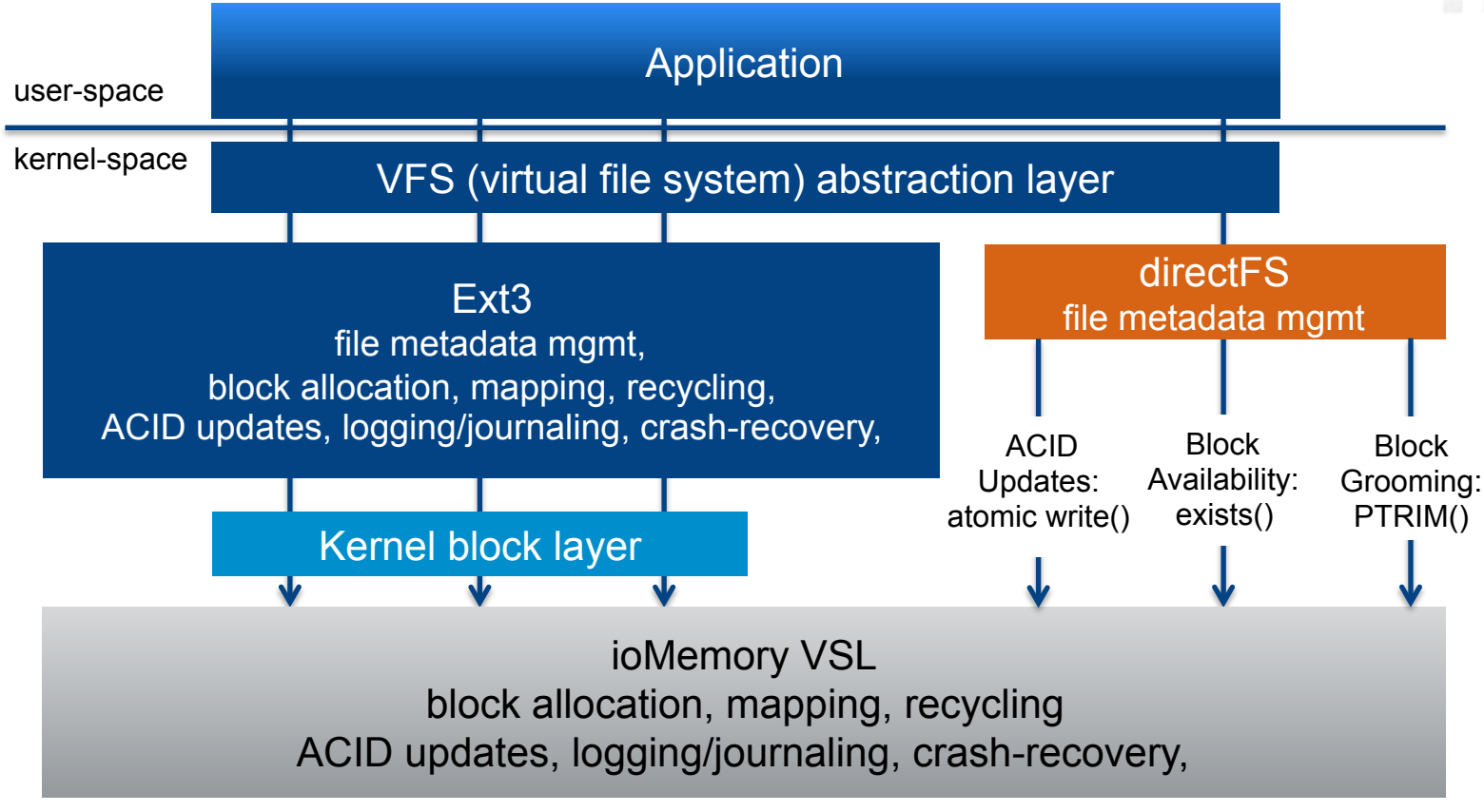
DIRECTFS

FUSION-IO

- Appears as any other file system in Linux
- Applications can use directFS filesystem unmodified with performance benefits
- Focuses only on file namespace
- Employ virtualized flash storage layer's logic for:
  - Large virtualized addressed space
  - Direct flash access
  - Crash recovery mechanisms
- Exposes VSL Primitives through file namespace
  - Applications can use SDK through directFS or directly to VSL

DIRECTFS – ELIMINATING DUPLICATE LOGIC

FUSION-IO



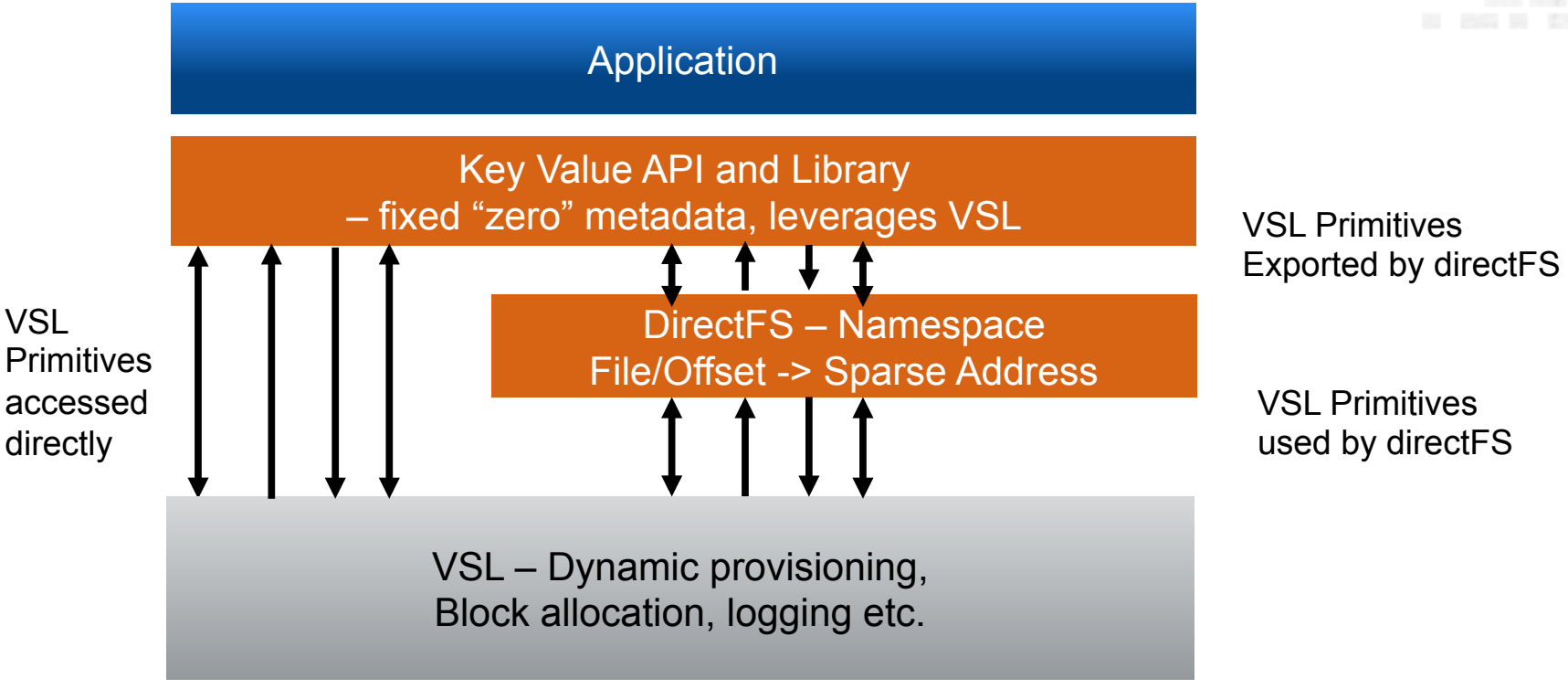
DIRECTFS SIMPLICITY – LINES OF CODE

FUSION-IO

File System	Lines of Code
DFS	6879
ReiserFS	19996
Ext4	25837
Btrfs	51925
XFS	63230

DIRECTFS – NATIVE FILE NAMESPACE FOR SDK API LIBRARIES

FUSION-IO

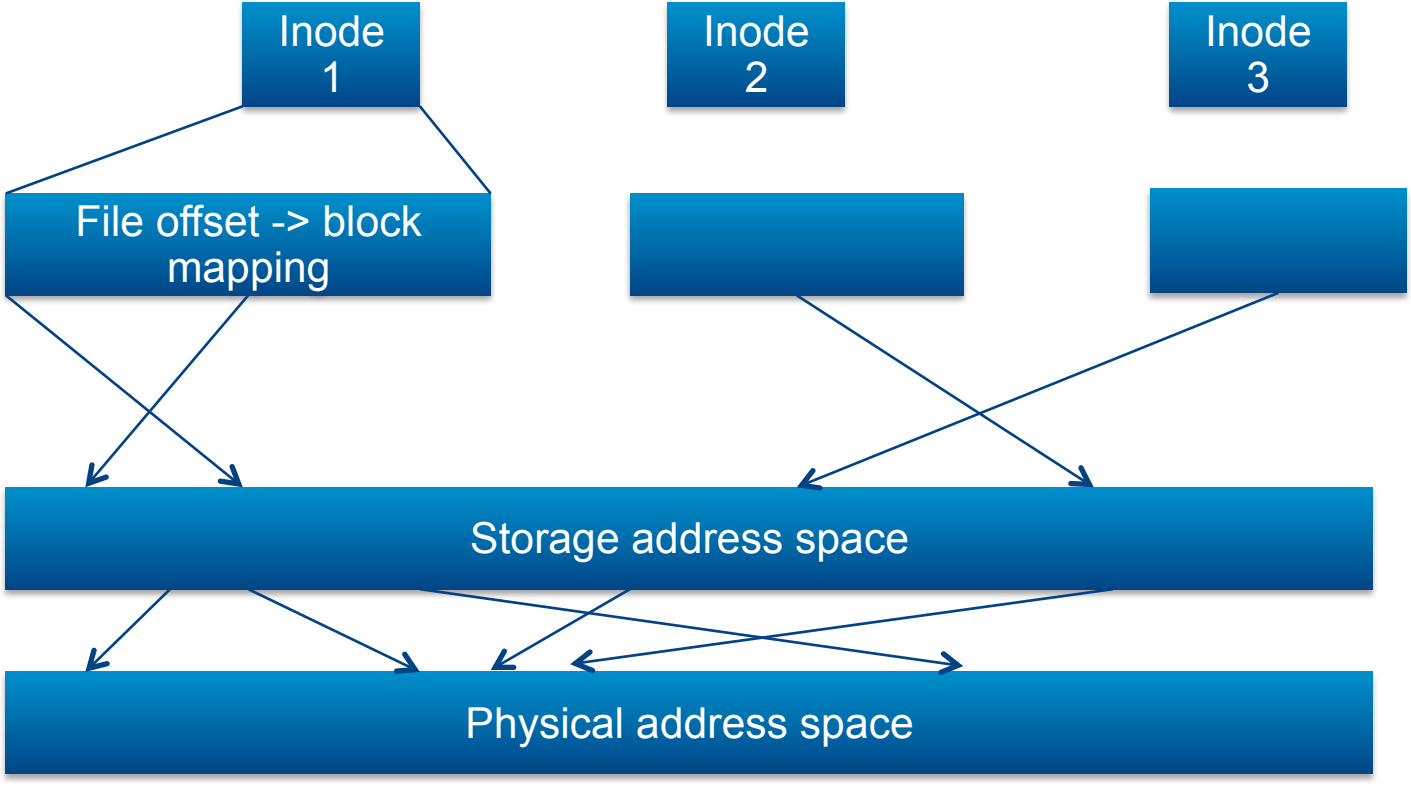


DirectFS exports VSL Primitives

DIRECTFS – SIMPLIFIED BLOCK MAPPING THROUGH NATIVE SPARSE VIRTUAL ADDRESSING

FUSION-IO

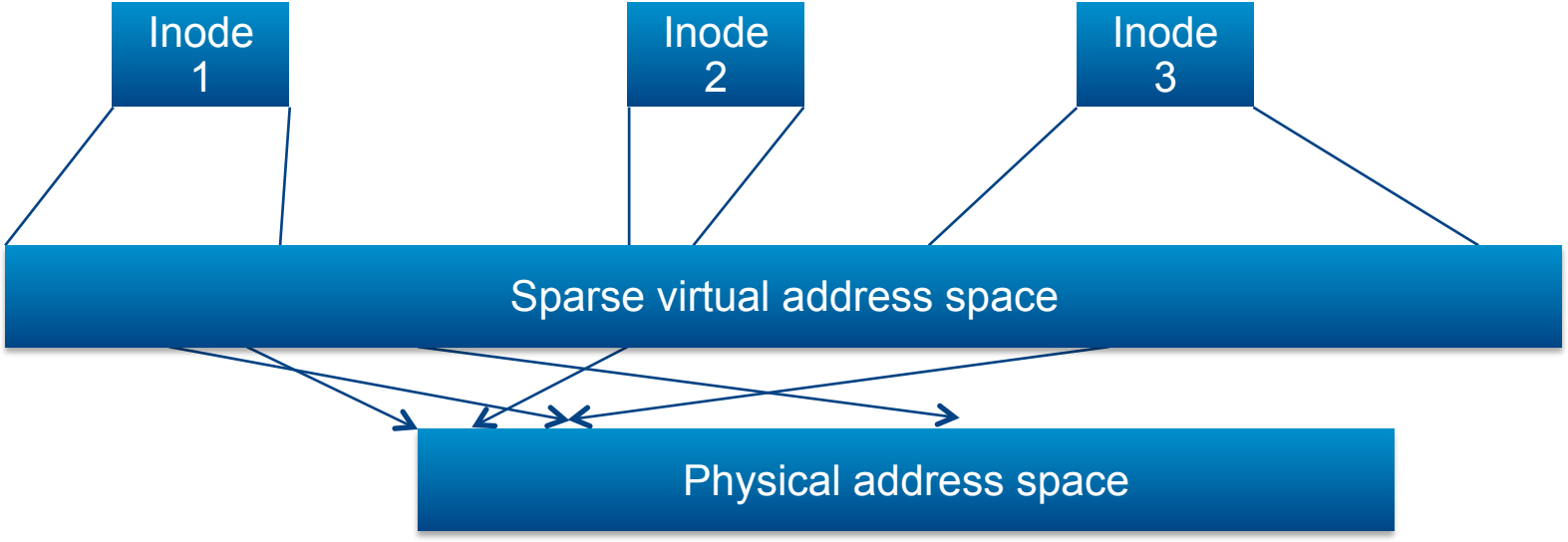
Conventional file systems on block storage



DIRECTFS – SIMPLIFIED BLOCK MAPPING THROUGH NATIVE SPARSE VIRTUAL ADDRESSING

FUSION-IO

- DirectFS – files mapped directly to sparse address space
- Removes mapping layers
- Allows file aware NVM optimizations and features



DIRECTFS – EXAMPLE: FALLOCATE

FUSION-IO

- Fallocate() – preallocate large files
- In conventional file systems
  - Write zeros for a large file to the physical media
- directFS
  - Pre-allocate virtual address space
  - Minimal writing to device – only to commit virtual address allocation
  - No need to write physical zeros to the entire file

DIRECTFS – CRASH SAFE THROUGH ATOMIC DATA AND METADATA UPDATE OPERATIONS

FUSION-IO

- Conventional file systems
  - Journal for power cut safety
  - Allows replay of transactions after crash
- directFS
  - Leverages atomics features of VSL
  - Can implement crash safety without journaling

DIRECTFS – INITIAL PROTOTYPE PERFORMANCE FOR UNMODIFIED APPS

FUSION-IO

Application	Ext3	DFS Prototype	Speedup
Quick Sort	1268	822	1.54
N-Gram	4718	1912	2.47
KNNImpute	303	248	1.22
VM Update	685	640	1.07
TPC-H	5059	4154	1.22


THANK YOU