Lecture17 Storage System

1. 瓦式磁盘

> Shingled Magnetic Recording (SMR) Disk

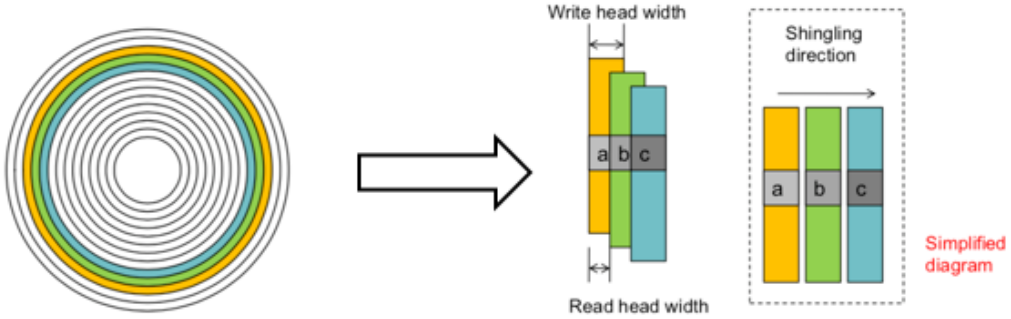
Overview

- 传统磁盘密度难以提升

  > 写磁头的宽度难以减小

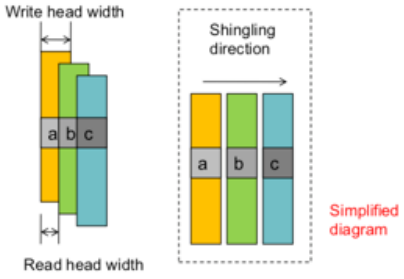
- 瓦式磁盘将磁道重叠，提升存储密度

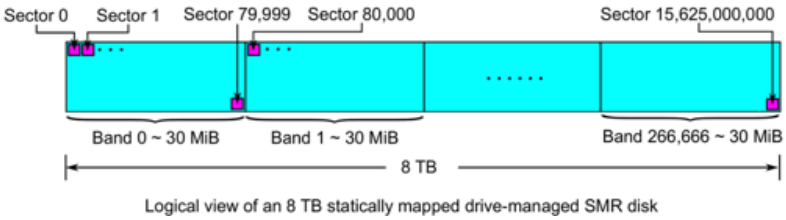
  > 减小读磁头的宽度



瓦式磁盘的问题：随机写

- 随机写会覆盖后面磁道的数据
  - 只能顺序写入
- 避免整个磁盘只能顺序写入
  - 磁盘划分成多个Band，Band间增大距离
  - 每个Band内必须顺序写入





Logical view of an 8 TB statically mapped drive-managed SMR disk

方法一：多次拷贝

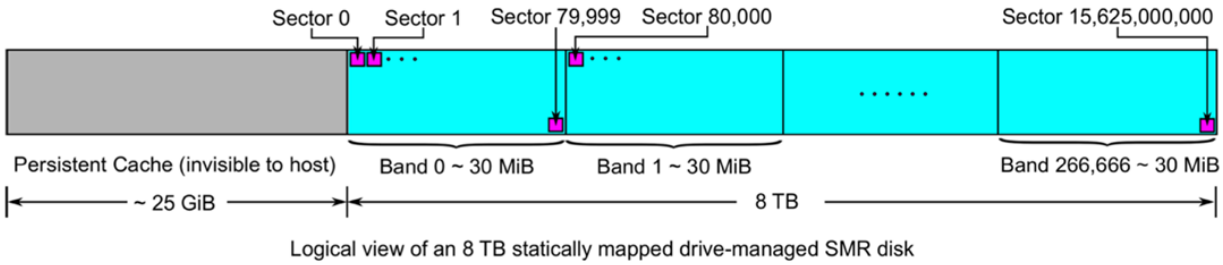
- 修改Band X的4KB数据
    1. 找到空闲的Band Y
    2. 从Band X的数据拷贝到Band Y，拷贝时将4KB修改写入
    3. 将Band Y中的数据拷贝回Band X

> 4KB随机写 -> 120MB访问

方法二：缓存+动态映射

- 大容量持久缓存
    - 在磁盘头部预留的区域，磁道不重叠，可随机写入
    - 给固件（STL）单独使用，外部不可见
- 动态映射：Shingle Translation Layer(STL)
    - 从外部（逻辑）地址到内部（物理）地址的映射
- 修改Band X中的4KB数据
    1. 将修改写入缓存，标记Band X为dirty
    2. 修改STL映射（让原位置指向持久化缓存）
    3. 空闲时，根据缓存内容，清理dirty Band

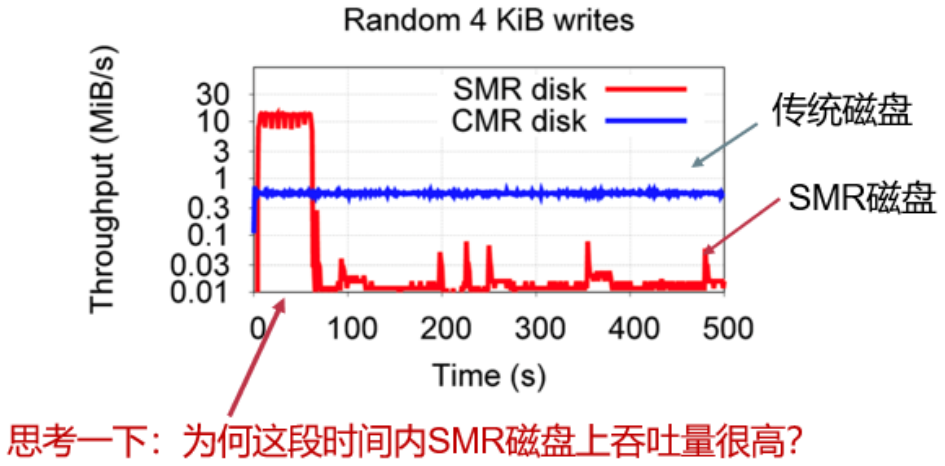
> 4KB随机写 -> 修改4KB缓存



Logical view of an 8 TB statically mapped drive-managed SMR disk

DM-SMR上使用Ext4

- 当随机写入时，Ext4吞吐量非常低!

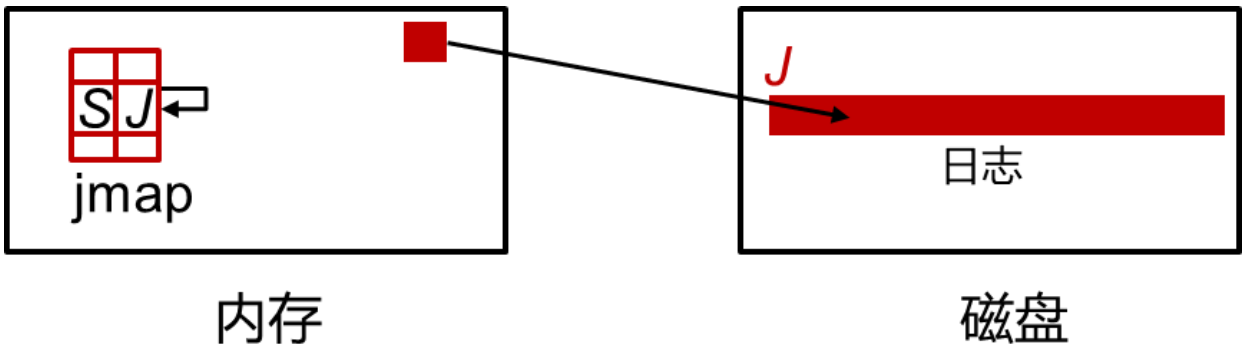


> 因为此时时写持久化缓存，STL将随机写映射为顺序写

解决方法：以LFS形式增加一个元数据缓存

- 以LFS形式维护10GB日志空间作为元数据缓存
  1. JBD2首先将元数据写入日志区域J，将元数据标记为clean(无需写回)
  2. JBD2在内存中的jmap中将S映射到J

  > 其中J表示日志区域，S表示脏元数据的应有位置
- Indirection：元数据访问需要通过jmap进行一次地址转换
- 日志空间的清理
  - 无效的元数据（被新修改过覆盖过的元数据）可以直接被回收
  - 对于冷的元数据，可将其写回到Ext4中其原本的位置S
  - 热的元数据继续保留在日志中
- 挂载FS时，读取日志，恢复出jmap

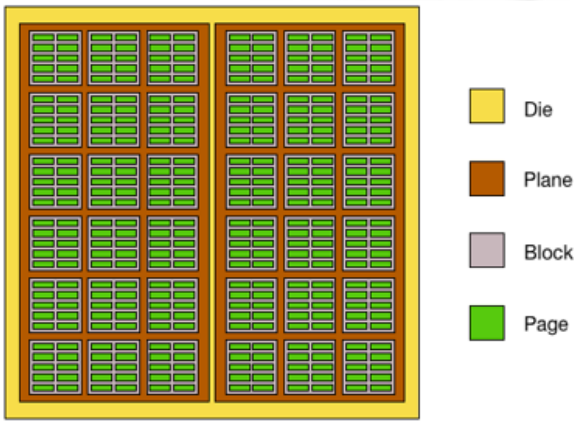


2. 闪存盘的文件系统

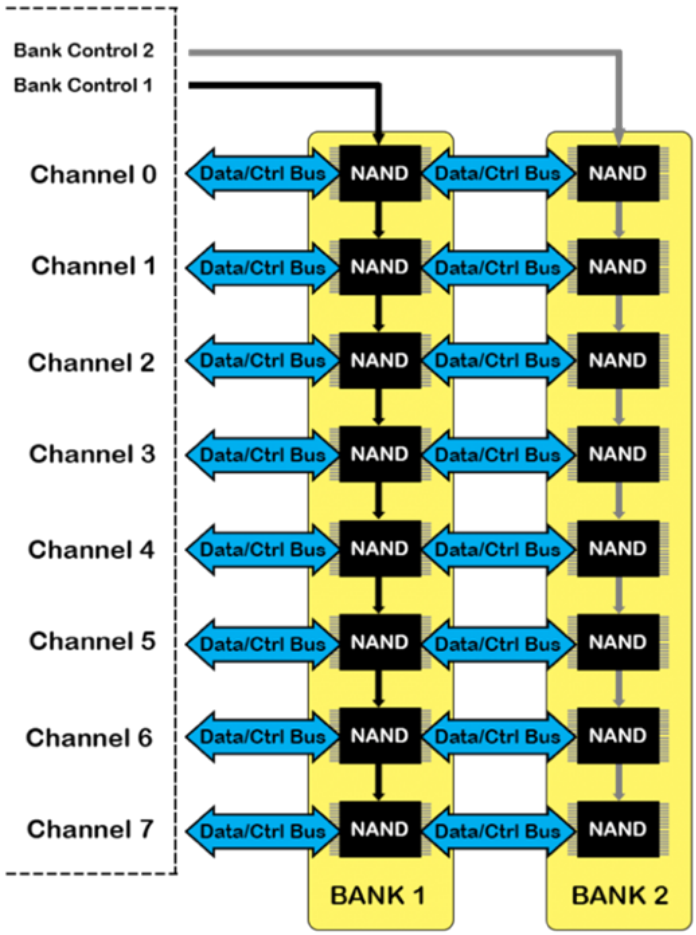
闪存盘的组织

闪存盘的组织

- (NAND) 闪存盘组织结构
  - A chip/package
    => 1/2/4 dies
    => 1/2 planes
    => n blocks （块）
    => n pages （页）
    => n cells
    => 1/2/3/4 levels



- 通道(Channel)
  - 控制器可以同时访问的闪存芯片数量（**同时读写**）
- 多通道(Multi-channel)
  - 低端盘有2或4个通道
  - 高端盘有8或10个通道



NAND Banks and Channels Illustration

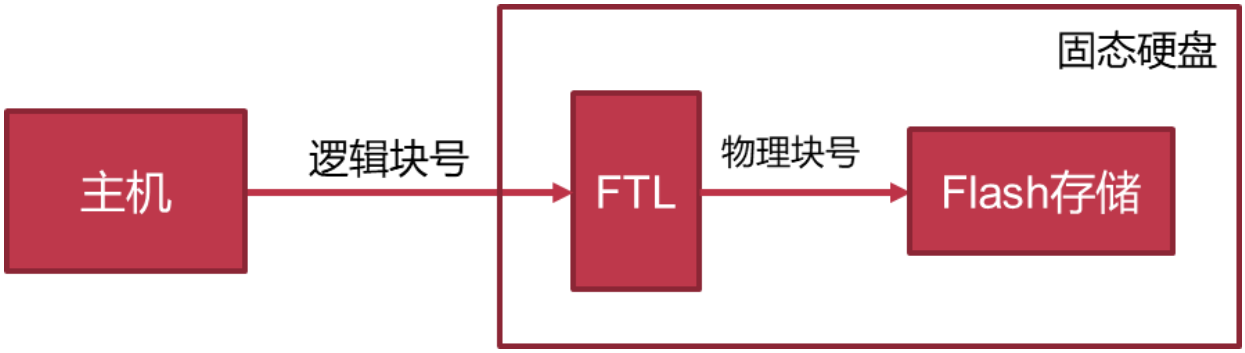
闪存盘的性质

- 非对称的读写与擦除操作
  - 页(page)是读写单元(8-16KB)
  - 块(block)是擦除单元(4-8MB)
- Program/Erase cycles
  - 写入前需要先曹处
  - 每个块被擦除的次数是有限的
- 随机访问性能
  - 没有寻道时间
  - 随机访问的速度提升，但仍与顺序访问有一定差距
- 磨损均衡
  - 频繁写入同一个块会造成写穿问题
  - 将写入操作均匀的分摊在整个设备
- 多通道
  - 高并行性

- 异质Cell
  - 存储1到4个byte：SLC、MLC、TLC、QLC

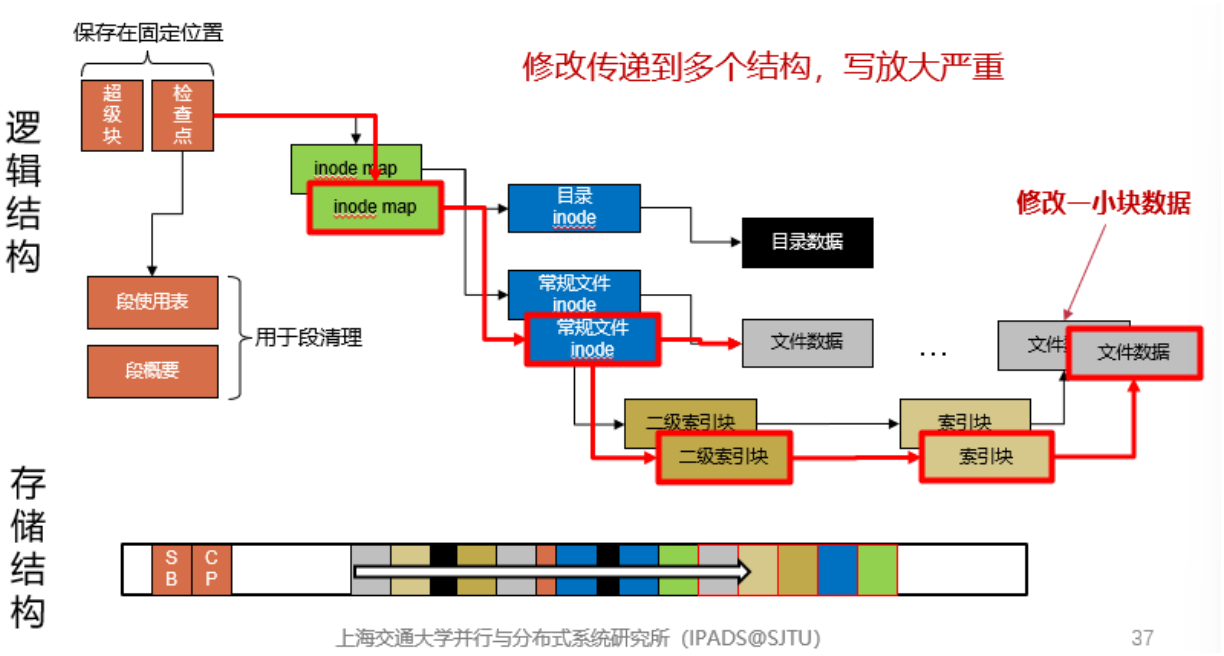
Flash Translation Layer(FTL)

- **逻辑地址到物理地址的转换**
  - 对外使用逻辑地址
  - 内部使用物理地址
  - 可软件实现，也可以固件实现
  - 用于垃圾回收、数据迁移、磨损均衡

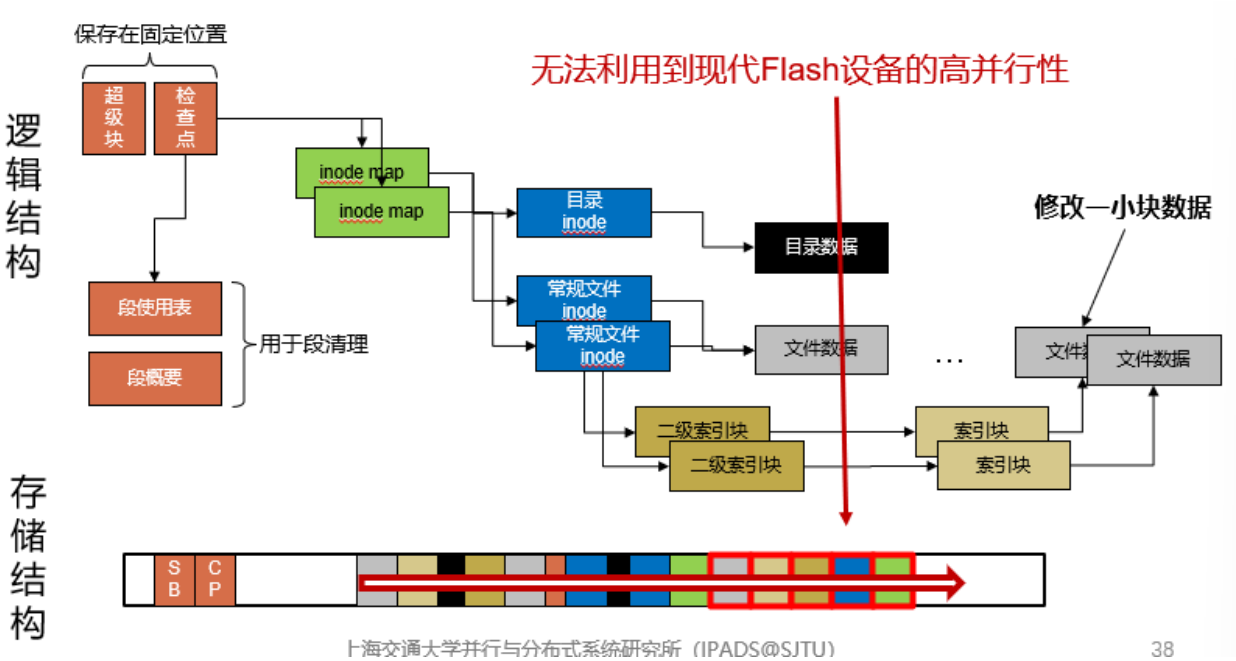


LFS的问题

- **递归更新问题**



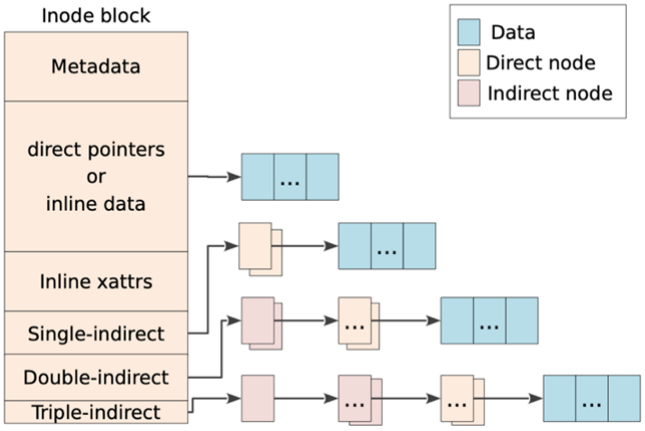
- **单一log顺序写入**

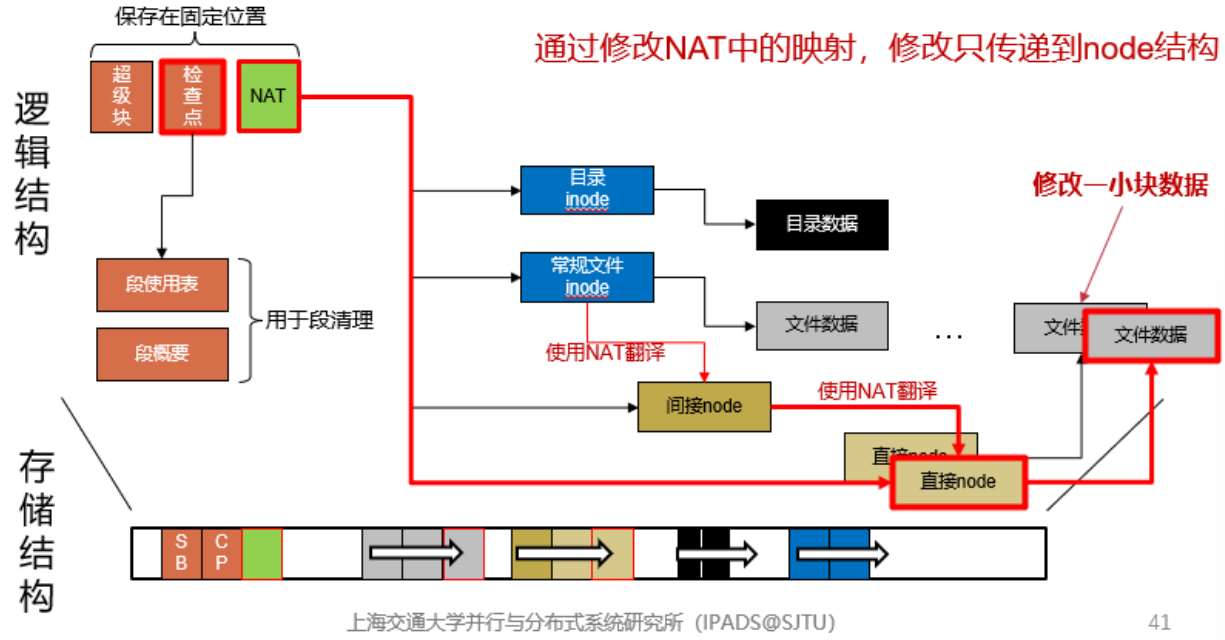


F2FS的改进

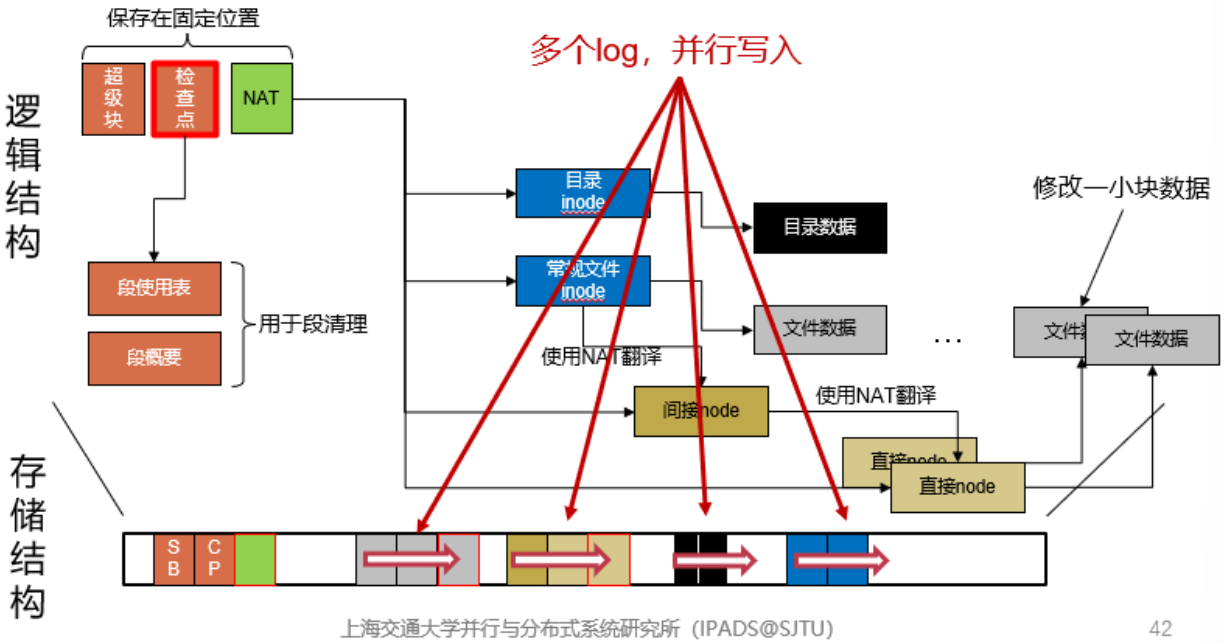
- **NAT**

1. 引入一层indirection：NAT(node地址转换表)
    - NAT：Node Address Table
    - 维护node号到逻辑块号的映射
    - Node号需要转换成逻辑块号才能使用
2. F2FS中的文件结构
    - 直接node：保存数据块的逻辑块号
    - 简介node：保存node号(相当于索引块)
    - 数据块：保存数据





- **多log并行写入**



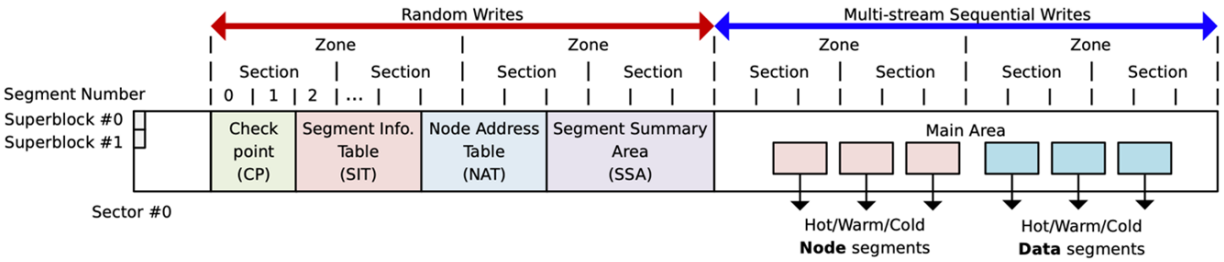
- **按热度将结构分类**
  - 每个类型和热度对应一个log
  - 默认打开6个log
  - 用户可进一步配置

- **根据硬件信息可以进一步调整**
  - 调整zone、section大小
  - 与硬件GC单元对齐等

类型	热度	对象
Node	热	目录的直接node块（包括inode块）
	温	常规文件的间接node块
	冷	间接node块
Data	热	存放目录项的数据块
	温	常规文件的数据块
	冷	被清理过程移动的数据块 用户指定的冷数据块 多媒体文件的数据块

闪存友好的磁盘布局

- 组织层级
  - Block：4KB，最小的读写单位
  - Segment：2MB
  - Section：多个segment（垃圾回收/GC粒度）
  - Zone：多个section
- 系统元数据（随机写入）
  - 存放在一起：局部性更好
  - CP：检查点
  - SIT：段信息表
  - NAT：node地址转换表
  - SSA：段概要区域
- 数据区（多Log顺序写入）
  - 区分冷/温/热数据
  - 区分文件数据(data segment)与元数据(node segment)

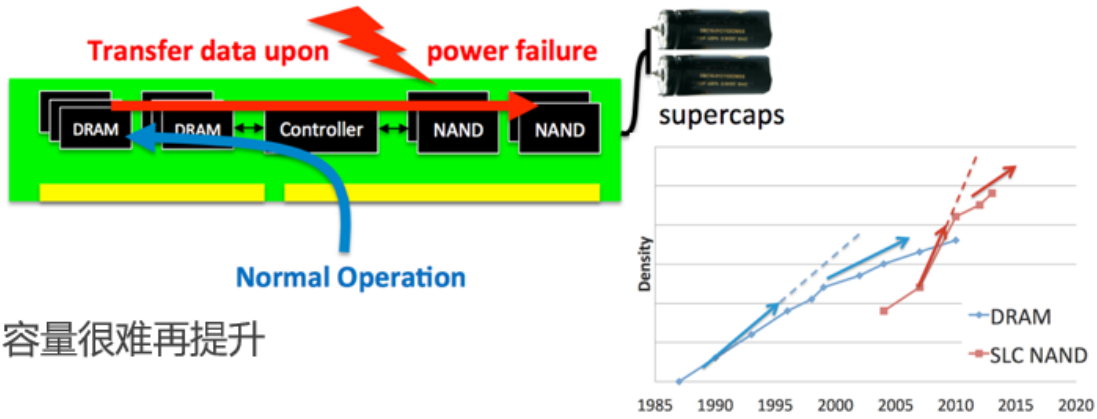


3. 非易失性内存

> Non-volatile Memory(NVM)

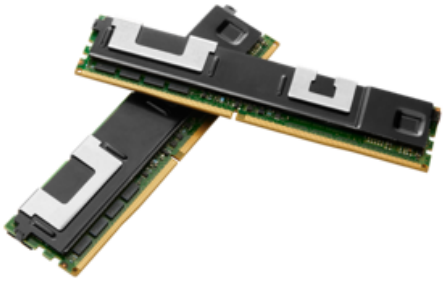
NVDIMM

- 在内存条上加上Flash和超级电容
  - 平时数据在DRAM中；断电后转移到Flash中持久保存



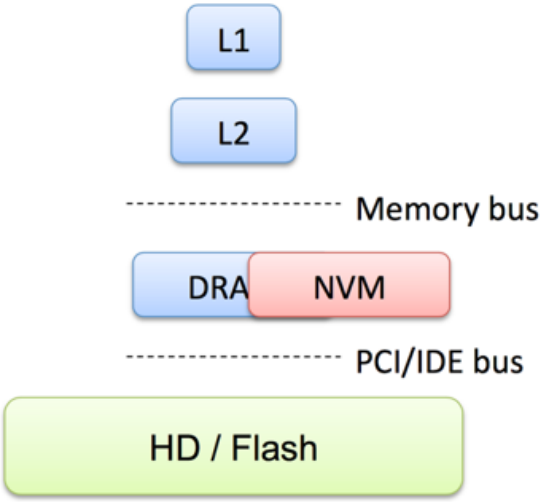
- 容量很难再提升

Intel Optane DC Persistent Memory



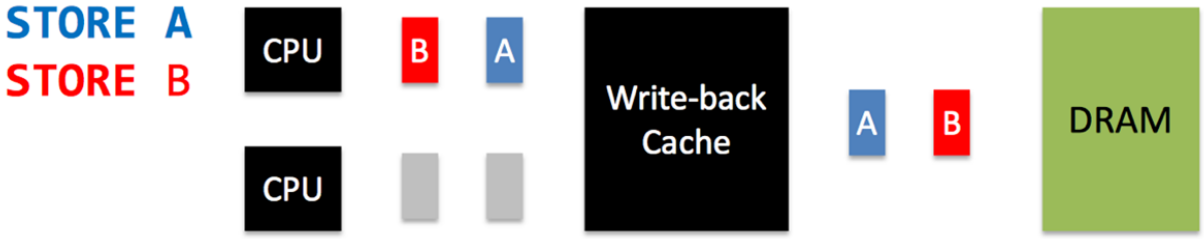
- ✓ 内存接口
- ✓ 字节寻址
- ✓ 持久保存数据
- ✓ 高密度 (512GB/DIMM)
- ✓ 需要磨损均衡，但耐磨度比NAND好10倍
- ✓ 比DRAM慢十倍以内，比NAND快1000倍

非易失性内存带来的新问题：内存写入顺序



	DISK	NVM
数据缓存位置	内存	CPU 缓存
写回	软件控制	硬件控制

- Writeback模式的CPU缓存
  - 虽然能提升性能，但会打乱数据写入内存的顺序



> 但考虑到持久性和一致性，写入的顺序十分重要

```
STORE data[0] = 0xFOOD
STORE data[1] = 0xBEEF
STORE valid = 1
```

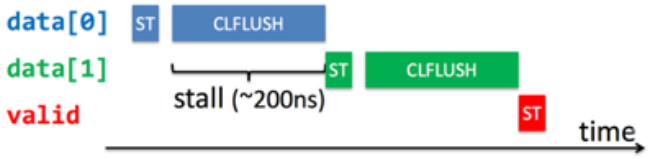

```
STORE data[0] = 0xFOOD
STORE data[1] = 0xBEEF
STORE valid = 1
```

使用CFLUSH保证顺序

- 使用CLFLUSH指令将数据逐出（Evict）缓存，以保证顺序

```
STORE data[0] = 0xFOOD
STORE data[1] = 0xBEEF
CLFLUSH data[0]
CLFLUSH data[1]
STORE valid = 1
```

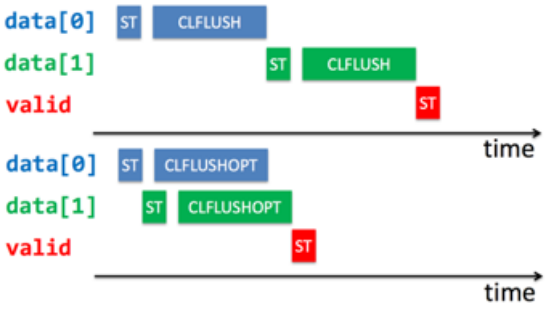
- CLFLUSH的缺点
  - 顺序执行，阻塞CPU流水线
  - 会将cacheline无效化（Cache-Line Flush的语义）



- 新指令：CLFLUSHOPT
  - 可以看做可并行执行的CLFLUSH
  - 需要用 sfence 来保证顺序

```
STORE data[0] = 0xFOOD
STORE data[1] = 0xBEEF
CLFLUSHOPT data[0]
CLFLUSHOPT data[1]
SFENCE // explicit ordering point
STORE valid = 1
```

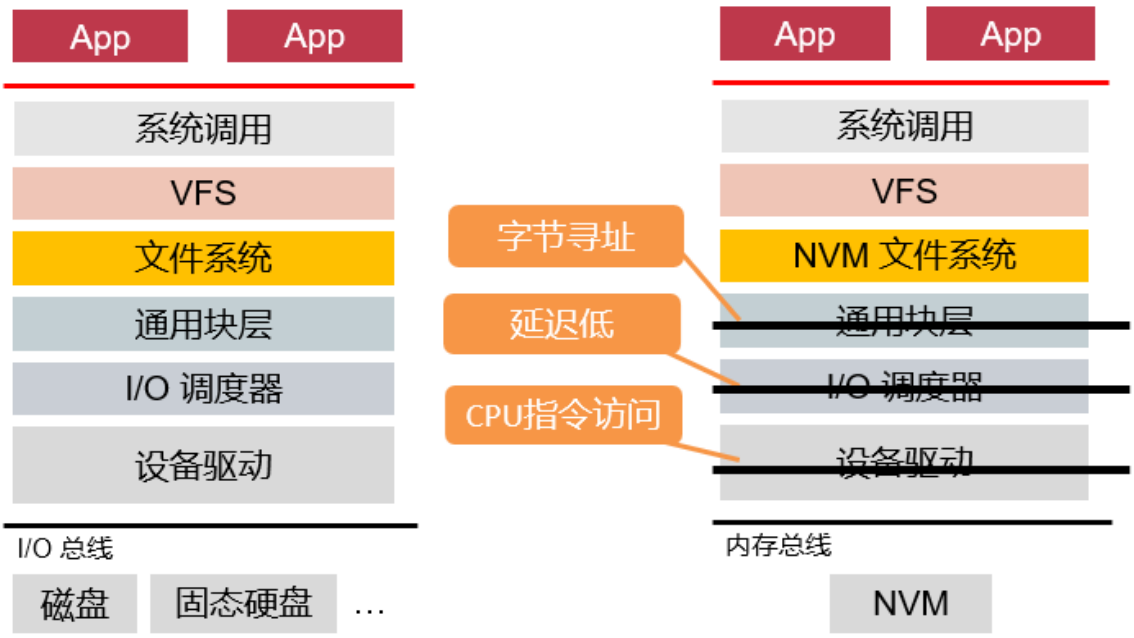
Implicit orderings



- 新指令： CLWB
  - Cache Line Write Back
  - 与CLFLUSHOPT类似，区别在于不会将cacheline无效化

4. 非易失性内存文件系统

非易失性内存改变存储栈



> 不想写了，感觉没什么意义...