Ssd原理

思路

- 由内向外的思路，先考察flash的原理，利用flash存储数据的基本单位：flash pkg，进而由flash pkg组成ssd，考察ssd的组成原理，ssd的使用特性，再向外扩展到ssd和内核的契合以及文件系统。
- 内容比较繁杂，都有点蜻蜓点水的特点。。。

agenda

- Flash技术
- SSD原理
- 内核体系结构相关
- 传统ssd与fusion-io对比

Flash技术

- 一种只能按照特大小的块擦除的EEPROM
- 包括NOR和NAND两种

Flash技术

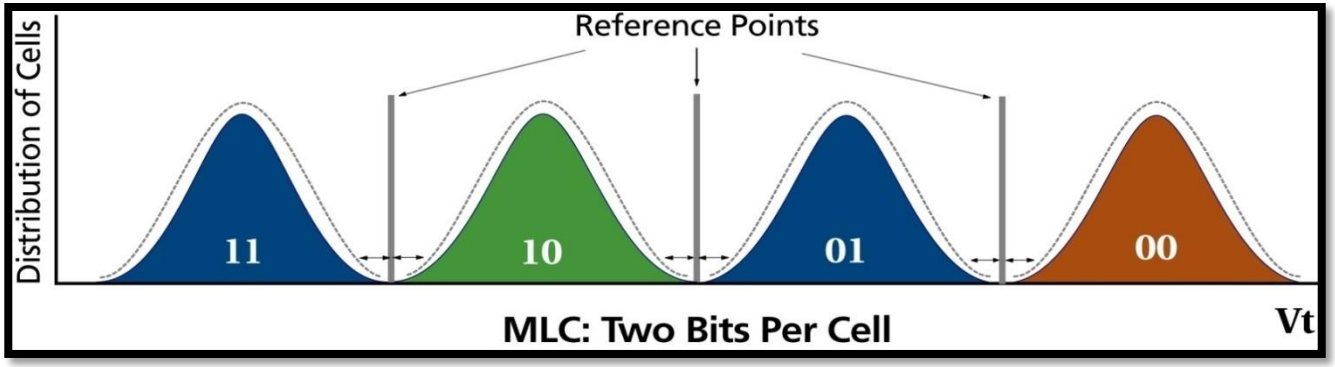
- NAND
- Flash chip体系结构

NAND

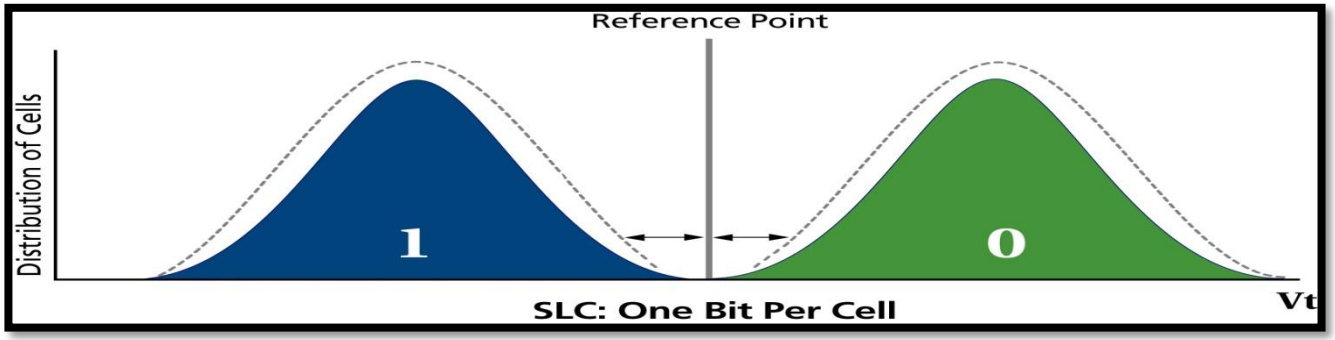
- Skip Principles of operation
- As a mass storage: MLC Vs. SLC
- Data operations: read program/write erase

What Is MLC NAND?

- MLC stands for multi-level cell NAND
  - MLC NAND stores 4 states per memory cell and allows 2 bits programmed/read per memory cell



- SLC NAND stores 2 states per memory cell and allows 1 bit programmed/read per memory cell



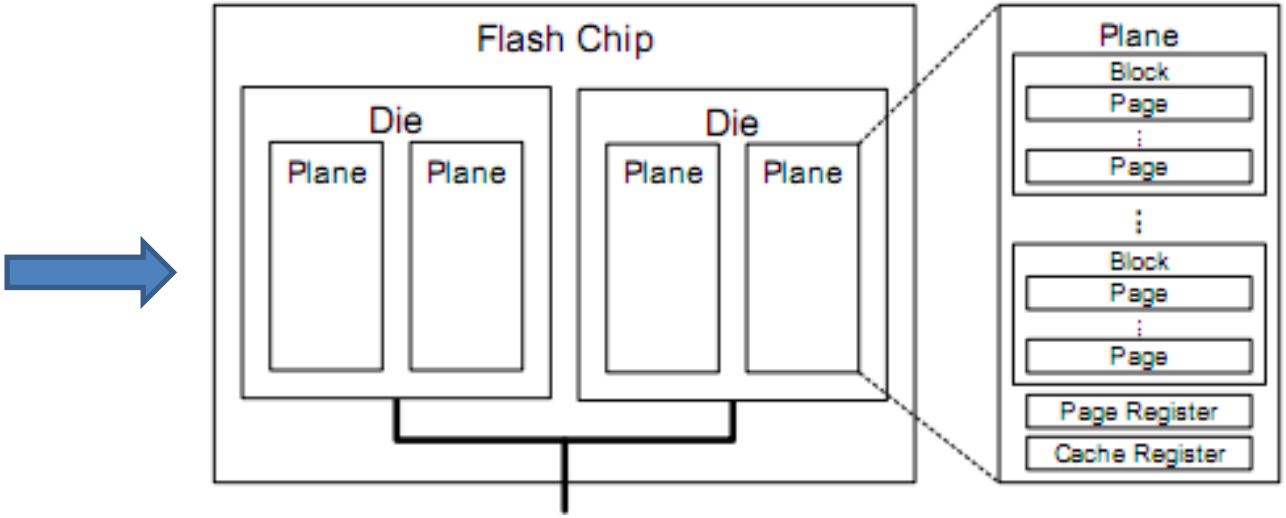
- SLC NAND具备两个电学状态，不同状态间有更大的间隔，因此具备更好的写入性能和数据保持能力;MLC NANS具备四个电学状态，每个结构单元有双倍的位密度，相对于SLC来说，在写入能力和纠错能力上较差，需要优化应用方法。

MLC versus SLC

Features		
Bits per cell	2	1
Voltage	3.3V	3.3V, 1.8V
Data width (bits)	x8	x8, x16
Architecture		
Number of planes	2	1 or 2
Page size	2,112– 4,314 bytes	2,112 bytes
Pages per block	128	64
Reliability		
NOP (partial page programming)	1	4
ECC (per 512 bytes) Error Correction Code	4+	1
Endurance (ERASE / PROGRAM cycles)	<10K	<100K
Array Operations		
tR (Max)	50µs	25µs
tPROG (Typ)	600–900µs	200–300µs
tBERS (Typ)	3ms	1.5–2ms

Flash chip体系结构

- NAND flash architecture was introduced by Toshiba in 1989.
- Vendors dependence
- Standardization (ONFI)



ONFI

- 开放式NAND闪存接口(ONFi)组织成立于2006年5月份，致力于推动NAND闪存接口的标准化，目前已有90多个成员。英特尔技术专家Amber·Huffman表示，目前ONFi组织目前的工作主要有两方面，从ONFi 3.0开始进一步提升NAND接口性能，从200MT/s到400MT/s，同时与ONFi 1.0和ONFi 2.0兼容;另外一个方面就是致力于零纠错码的NAND发展。将依赖于NAND制程的功能从控制器中剥离出来，以减轻主控的负担，使得NAND性能得到进一步提升。
- http://storage.chinabyte.com/26/11220526_4.shtml

Flash chip体系结构

- Within a die planes can be interleaved only when the same types of commands are issued together.
- A die is an independent unit that has its own ready/busy and chip-enable signals.
- The dies in a chip can execute different operations from each other in an interleaved manner.
- 每一个plane都由一定的block组成，block又由一定数量的page组成。

Data operations

- 操作
  - 读 从page中读取数据到寄存器。
  - 写/编程 将寄存器内的数据编程进可编程page
  - 擦除 读和写都是以page为单位的，擦除是以block为单位进行操作，把所有的cell都变为1。

Sample

- 读速度：

  读取一个page的时间是Page read to register + serial access to register = 125us，所以读取速度为10e6/125*4e3 = 32Mbytes/s

- 写速度（理想，不考虑擦除）：

  200us+100us = 300us，速度为

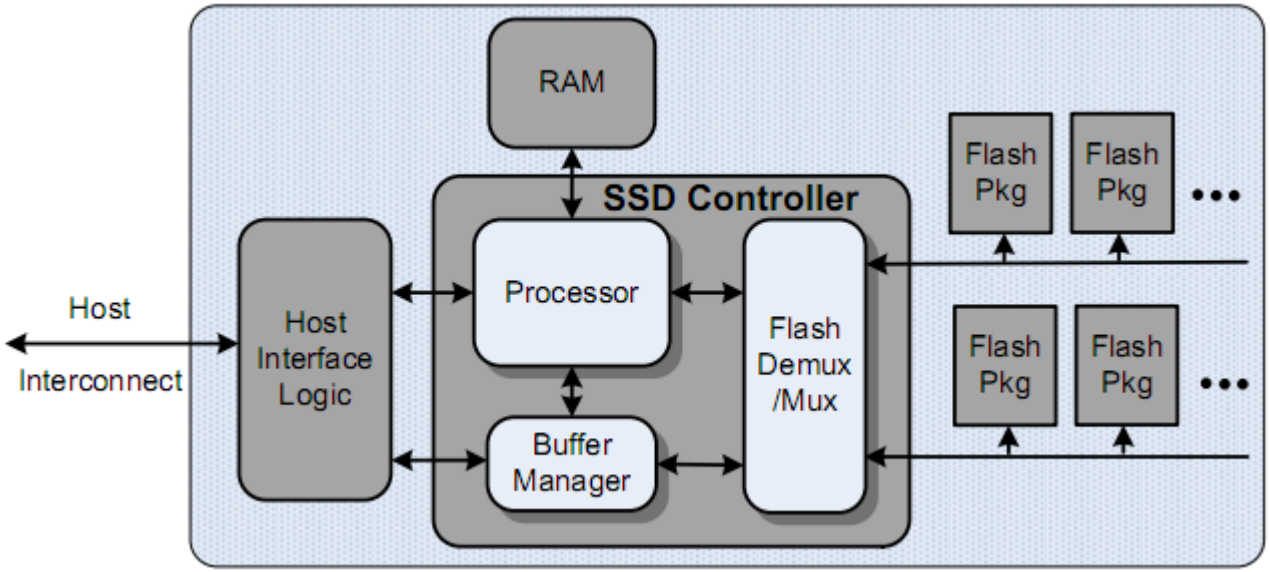
  10e6/300*4e3 = 13Mbytes/s

Page Read to Register	25μs
Page Program (Write) from Register	200μs
Block Erase	1.5ms
Serial Access to Register (Data bus)	100μs
Die Size	2 GB
Block Size	256 KB
Page Size	4 KB
Data Register	4 KB
Planes per die	4
Dies per package (2GB/4GB/8GB)	1,2 or 4
Program/Erase Cycles	100 K

- Feature
  - Low power consumption
  - Low noisy
  - Fast IOPS and fast io rate
  - Reliability
  - Weight & shock & low temperature
  - More expensive per bit

SSD原理——物理构件

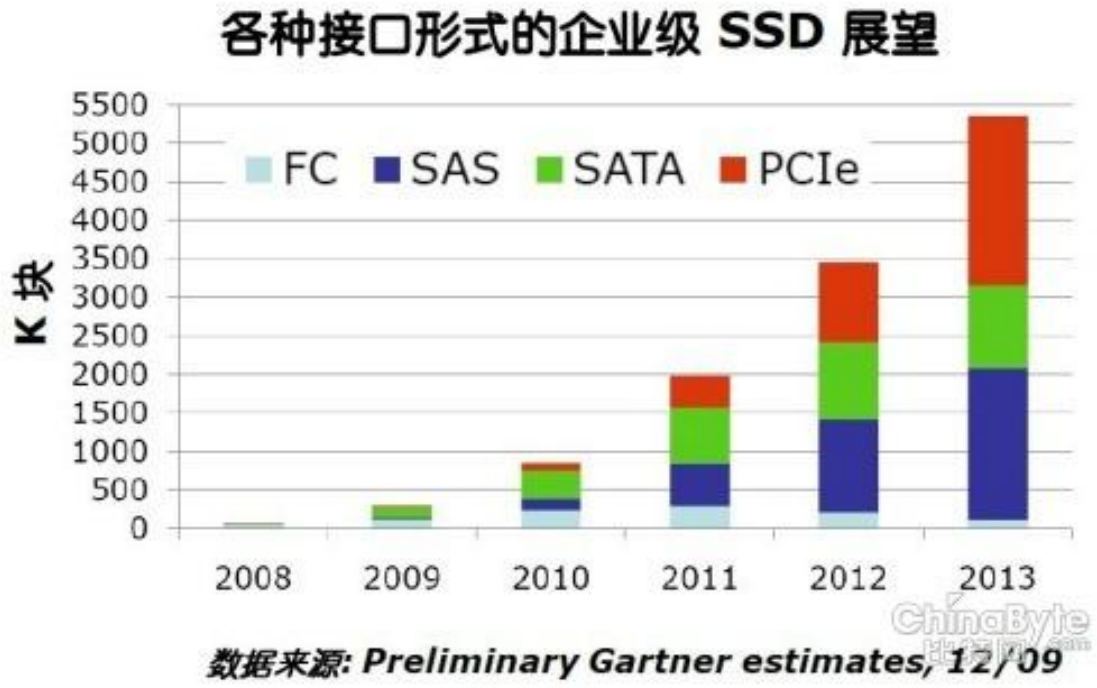
- Processor
- Interface
- Flash Mux
- Buffer
- Flash Pkg



通常controller使用FPGA硬件实现，所以处理速度是非常快的，接口可以使SATA、PCIE等等。每一个flash pkg都有8位的串口数据输出，当容量增大的时候需要使用Flash Mux进行转换。

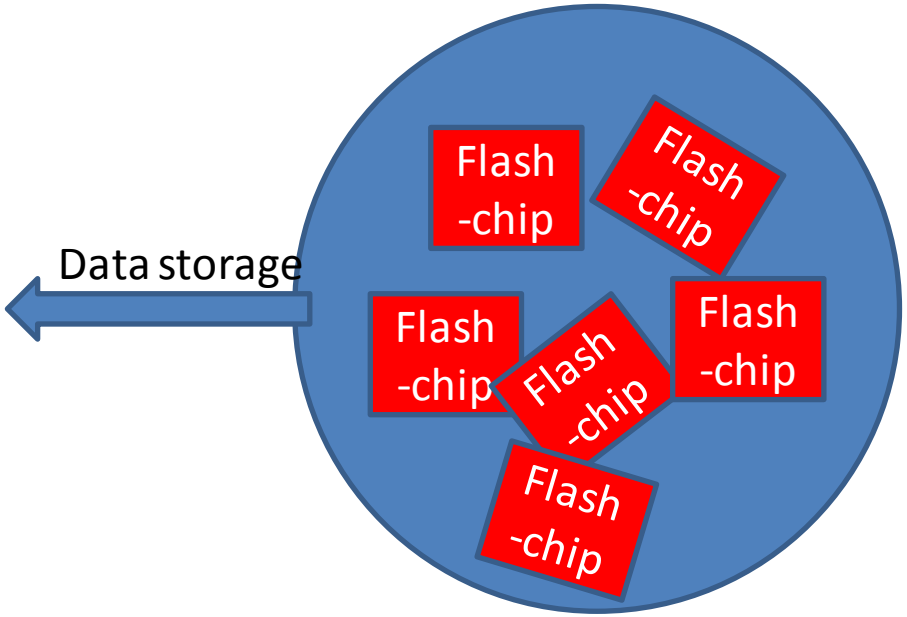
Interface

- 除了传统接口，英特尔认为需要制定企业级的NVMHCI(非易失性存储主机控制器接口，Non-Volatile Memory Host Controller Interface，简称NVMHCI)标准。



SSD原理——使用逻辑

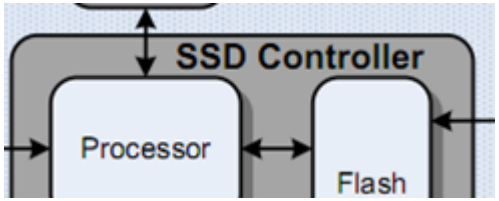
- 巨大的二进制存储空间flash-pkg，如何组织这些空间达到存储非遗失数据的目的？
  - Ssd内部设计
- 如何和现有操作系统结合？
  - 驱动接口



SSD原理——使用逻辑

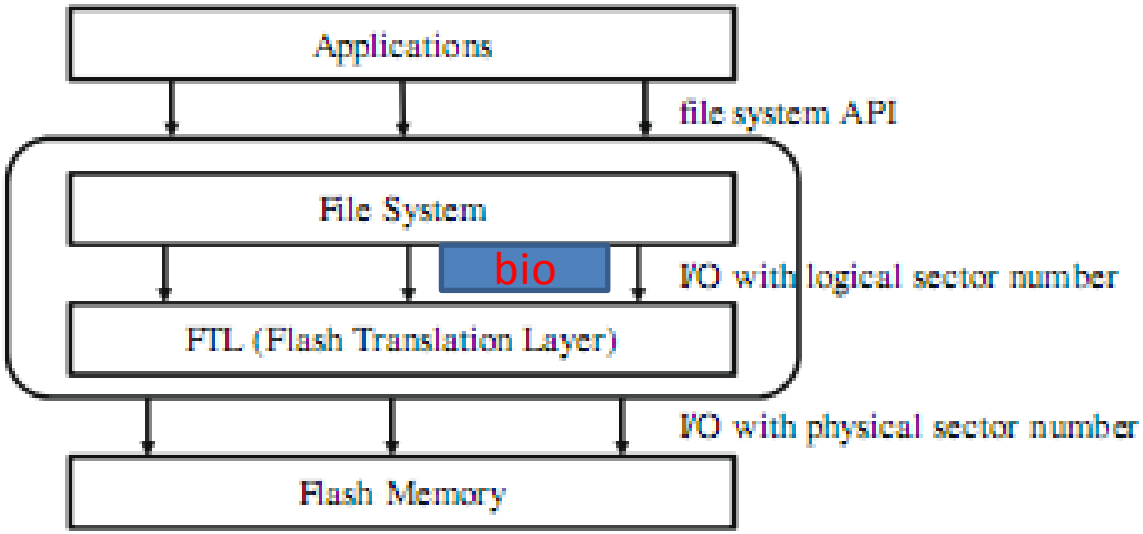
- Allocation pool

  When handling a write request, each target logical page (4KB) is allocated from a pre-determined pool of flash memory. The scope of an allocation pool might be as small as a flash plane or as large as multiple flash packages.



Recall-构造通用的块IO请求

- 表示一次磁盘连续IO操作的数据结构为bio，bio在通用块层构造，构造好之后传递给I/O调度层，进而传递给块驱动层来处理。磁盘操作的最小单位是扇区，所以bio中使用起始扇区号和长度来表示这次IO操作（读or写）所涉及的物理区域。



SSD原理——FTL

- FTL

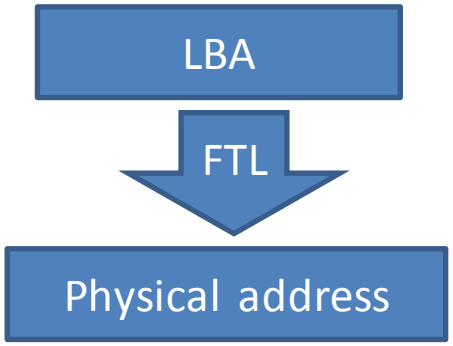
  performs the virtual-to-physical address translations and hides the erase-before-write characteristics of flash.

FTL

- 功能
  - Logic block map
    - 块设备控制器都需要的一种功能。
  - Reclamation
    - 由于flash的写前需要擦除以及读写单位为page，擦除单位为block的特点决定。
  - Wear leveling
    - 由flash芯片的P/E次数限制，也就是flash chip的使用寿命限制。

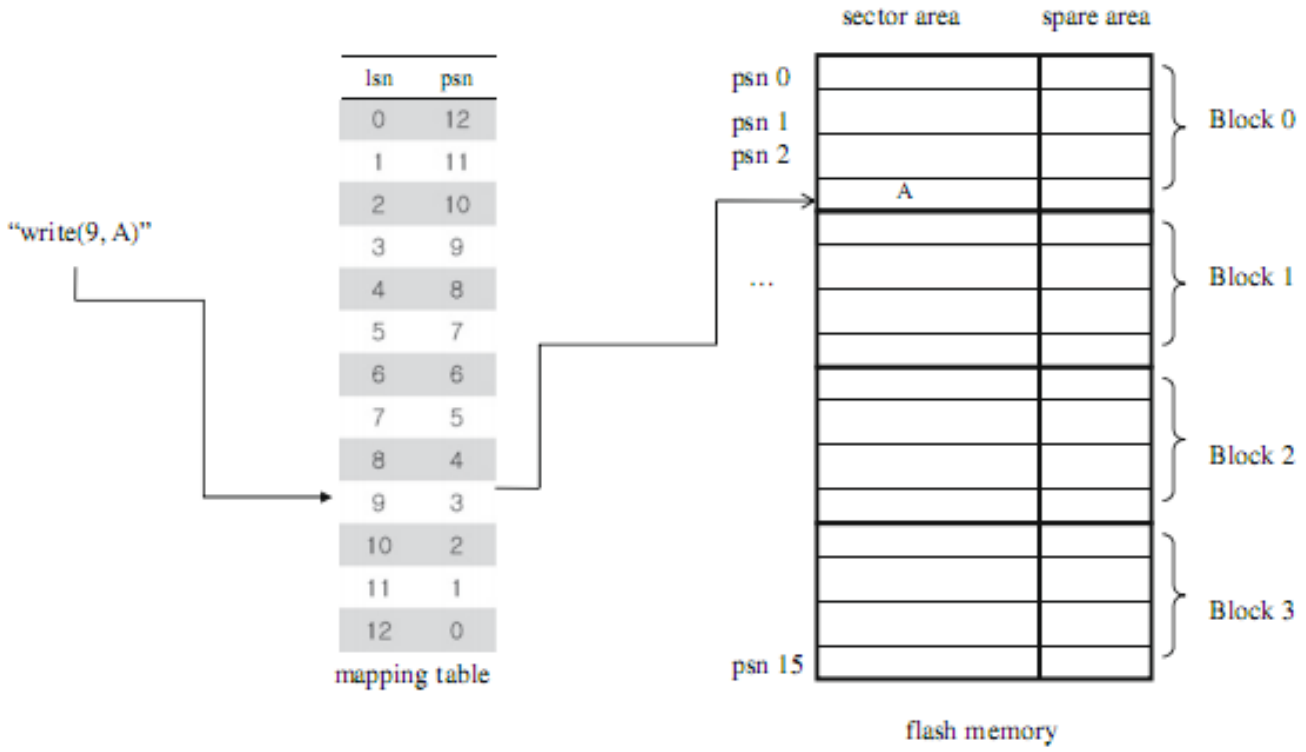
Logic block map

- page mapping
- Block mapping
- Hybrid mapping



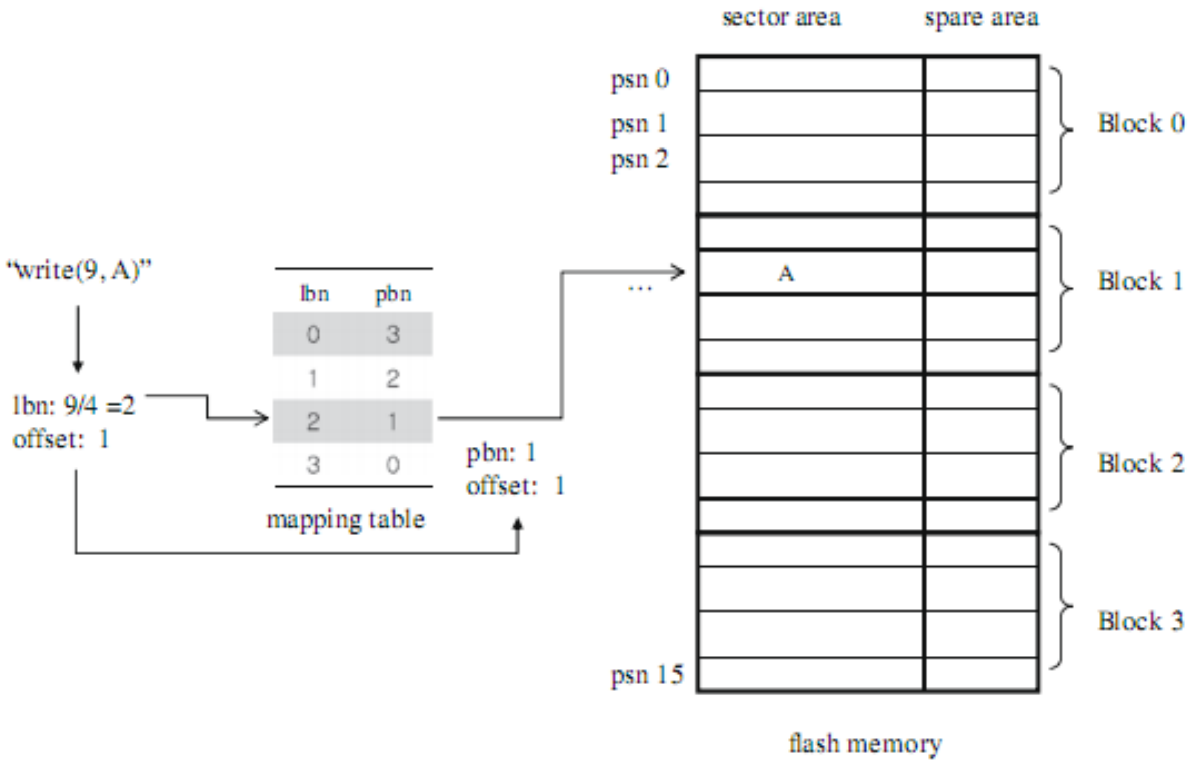
page mapping

- 在mapping table中，每一个LBA，都映射到某一个PGA，所以要找到某一个LBA中的内容只需要查找mapping table即可。



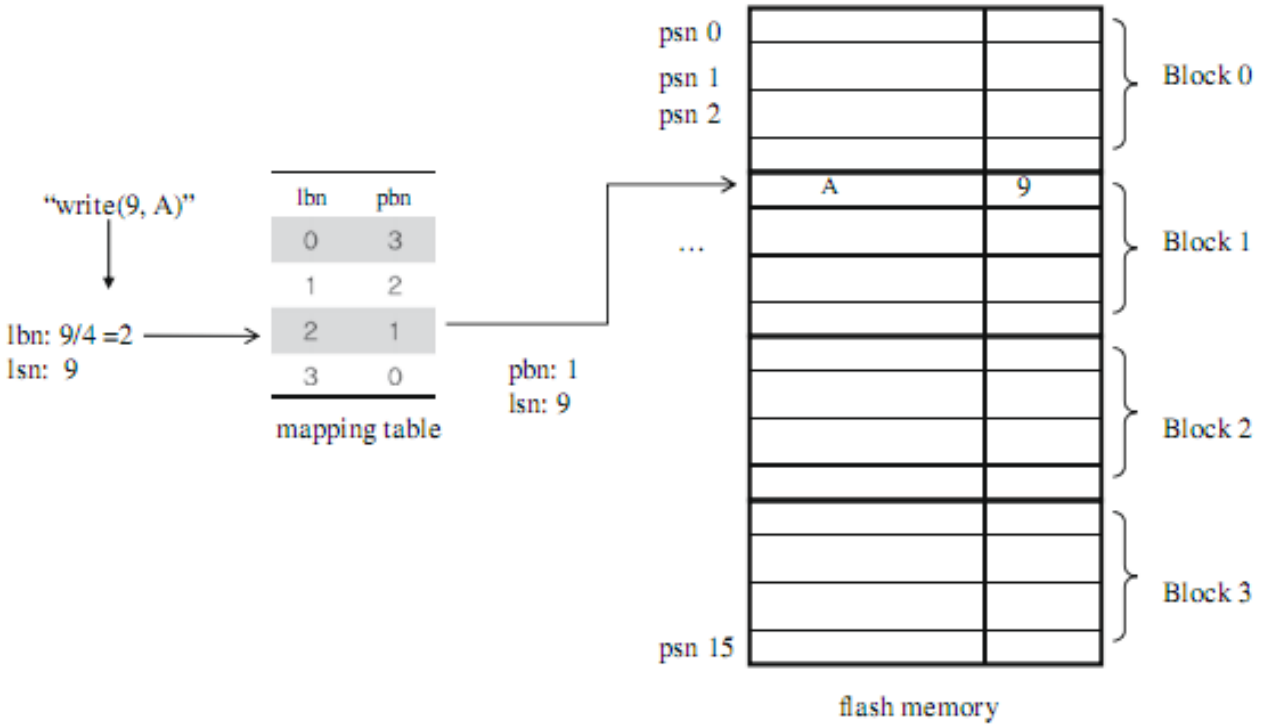
Block mapping

- 类似于cache中的直接映射，LBA被分为两个部分，前一部分被用作索引block号，后一部分索引block内的page号。



Hybrid mapping

- 类似cache中的set association，LBA被放置到每个page的隐藏数据区，需要读取某一个LBA的数据时，需要读取整个block的每一个page中的隐藏数据区的值进行比对，直到找到该LBA。



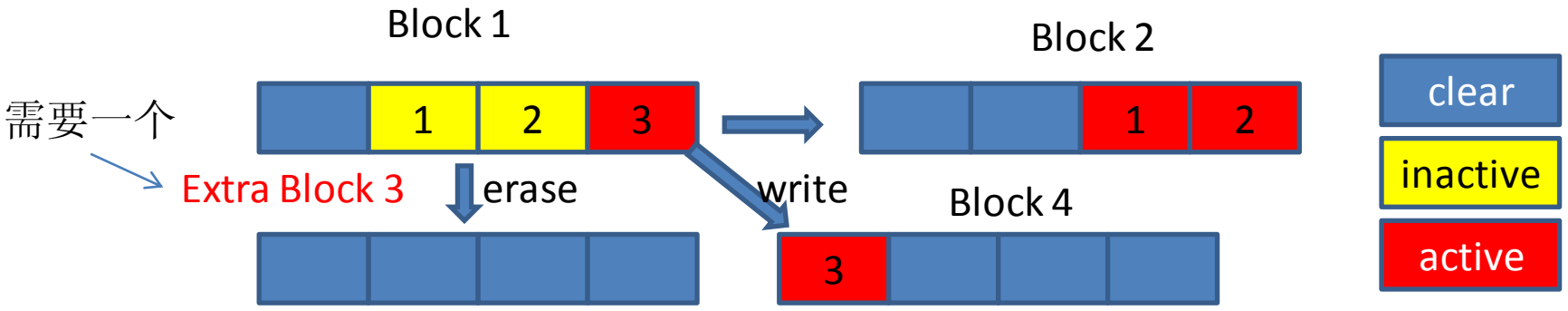
对比

- Page mapping需要大的ram，但灵活性高，可以做损耗均衡等等。
- Block mapping的ram需求量小，但是会导致性能的下降，不灵活。
- 混合映射即使一种折中，但是读取某个page需要读取block中的page进行LBA的比较。

Reclamation

- **Cleaning：** 当逻辑页（文件系统block的概念）的大小小于flash的block的时候，写入一个逻辑页的数据就不会占满整个flash的block，所以就会出现：某一个block包含了空闲page，已使用但未过期的page，已使用已过期的page，所以为了以后的页能够写入，需要回收已使用已过期的page。

两个时刻的转换，时刻1，写入了3个逻辑页，时刻2又从新写入1、2，导致Block 1，中的黄色页过期，所以需要回收。



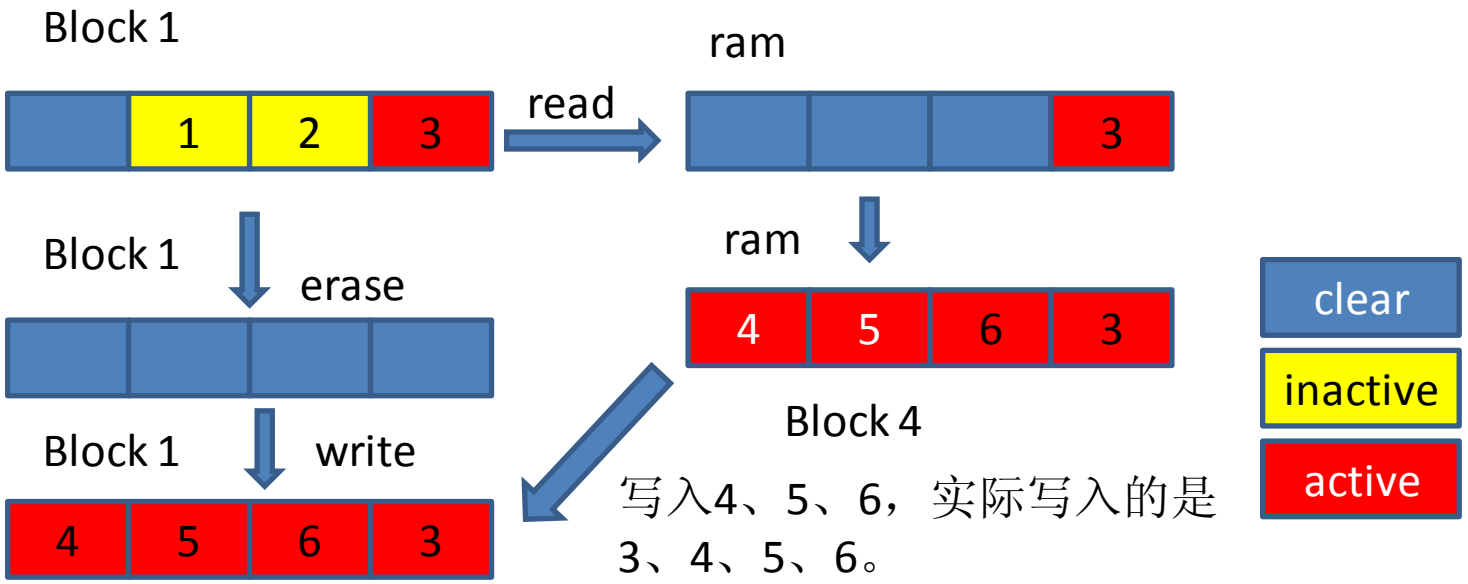
回收需要一次写的操作，一次擦除操作，而回收的往往是由写操作（导致页的过期）引发的。

Reclamation algorithm

- 回收算法的好坏直接影响ssd的性能，文献中多使用贪心算法，或者使用一定大小的ram来优化。比如，记录block中的无效数据page的个数，选择无效数据多的进行回收。也可以为了保持ssd整体损耗和负载的均衡，选择使用次数少的进行回收等等。

Write amplification

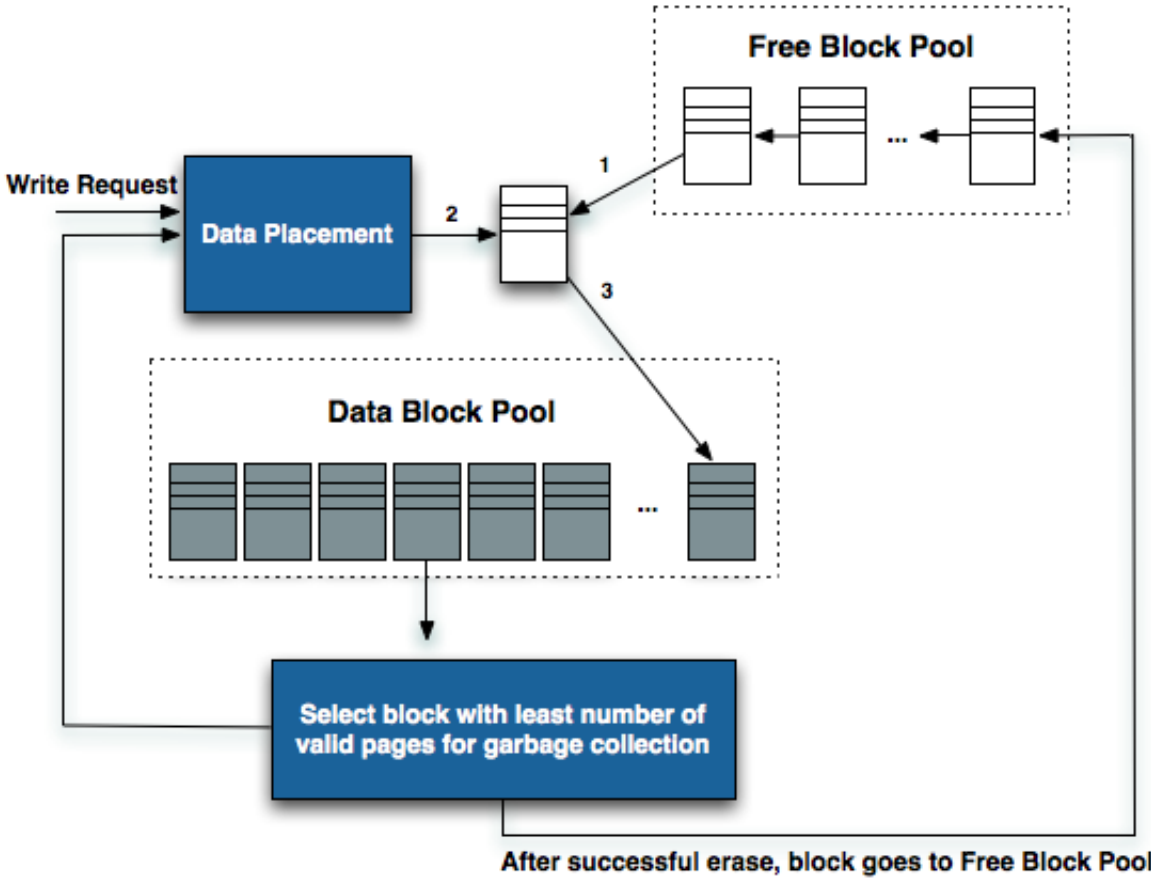
- 由flash的写特性导致（脏页写前必须擦除），需要先将block中的有效page读出到ram中，然后擦除block，最后加上新的数据写入block。



Spare area—减少写放大

- 为了优化写的性能，将一些block分成一个pool（spare area），做clean时，由于write可以在flash内部执行，因此，可以提升写性能，减少写放大。

大的spare area
意味着大的碎片
整理能力，因为有
更多的空间可以被用来
存放临时性的东西。



图上有链接

Spare area

- 由于wear out的存在，写放大越小，ssd的寿命越长，而spare area可以减小写的放大，并且可以做更多wear leveling的优化，企业级的ssd都有分配一定的block给spare area。

SandForce Overprovisioning Comparison			
Advertised Capacity	**Total Flash**	**Formatted Capacity (28% OP)**	**Formatted Capacity (13% OP)**
50GB	64GB	46.6GB	55.9GB
100GB	128GB	93.1GB	111.8GB
200GB	256GB	186.3GB	223.5GB
400GB	512GB	372.5GB	447.0GB

Wear leveling

- 做损耗均衡就是将write尽可能的分布到所有的block，从而增大ssd的使用寿命。
- 相关性
  - Mapping algorithm 映射算法直接影响损耗均衡，如果采用大的allocation pool，则可以使得负载比较均衡
  - Cleaning（reclaiming）比如clean选择时尽可能考虑在某一个使用次数threshold以上的blcok（具体算法不同）。
  - Cold data & hot data 两种数据分离存放有助于使用寿命的延长，否则，cold data和hot data混合的block回收时将会产生没有必要的write。（参考：**The SSD Relapse: Understanding and Choosing the Best SSD**）

内部结构对性能的影响

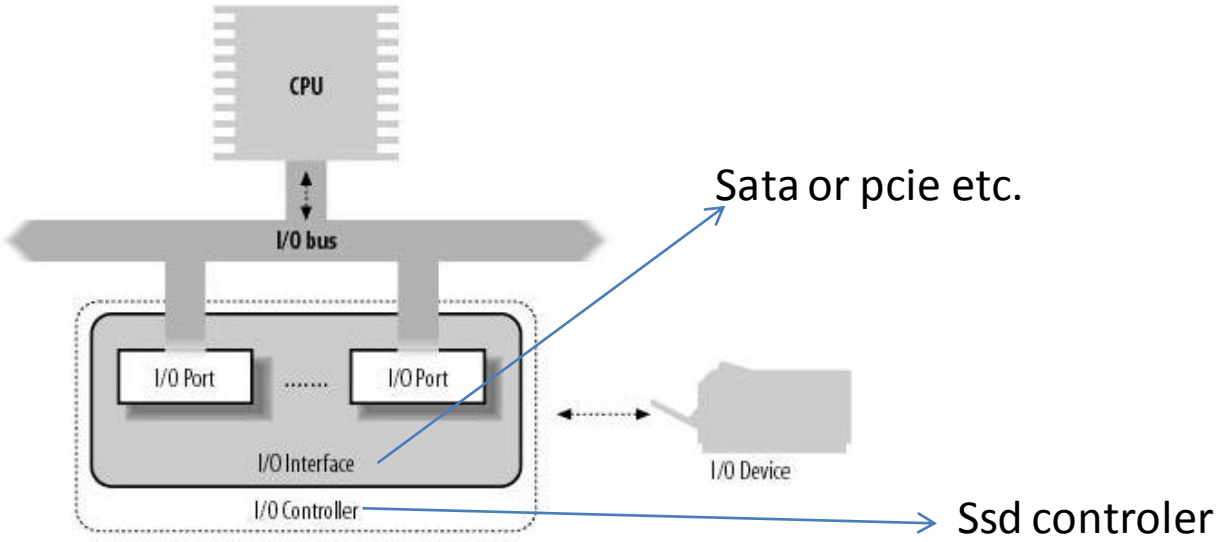
- 大的分配池
  - 更好的负载平衡。
- 大的page size
  - 小的映射表，但是当逻辑页小于page size时，会产生Read-modify-writes。
- 更多的spare area
  - 减少clean的次数，因为大的spare area可以是的block中的有效数据比例更大。但是也减少了可用空间。
- 数据条带化，增加读取并行度
  - 使得数据分布到不同的flash pkg中，增加并行度。

SSD在服务器上的应用

- 与桌面对比
  - 更高的读写性能、稳定性等等。
- 与嵌入式对比
  - 嵌入式的flash设备多数是裸的，没有FTL，wear leveling等都是基于文件系统的。
- 优化
  - FTL的优化，根据读写特性，有的提出大量写的优化，有的提出加速读写的优化，都是针对SSD内部控制器的优化。
  - 完全抛掉FTL的优化。

内核体系结构相关

- 内核驱动架构



- 对于内核来说ssd就是一个普通的块设备文件，和普通磁盘没有任何区别，ssd中的FTL层，隐藏了ssd的读写特性，并在ssd内部实现了通用的io接口。

文件系统

- 基于有FTL的flash设备（ssd等）
  - 传统文件系统ext等，为对ssd进行任何优化
  - Btrfs等现代文件系统，可以开启对ssd的优化
- 嵌入式上面的针对裸flash设备
  - yaffs、jffs、logfs等
- Log-structure文件系统更加适合flash设备。

MTD

- 为了更好的使用flash设备，提出了一种新的设备文件MTD（linux有六种文件），新的使用接口UBI，以及在此接口基础上实现的一个文件系统ubifs。

传统ssd与fusion-io

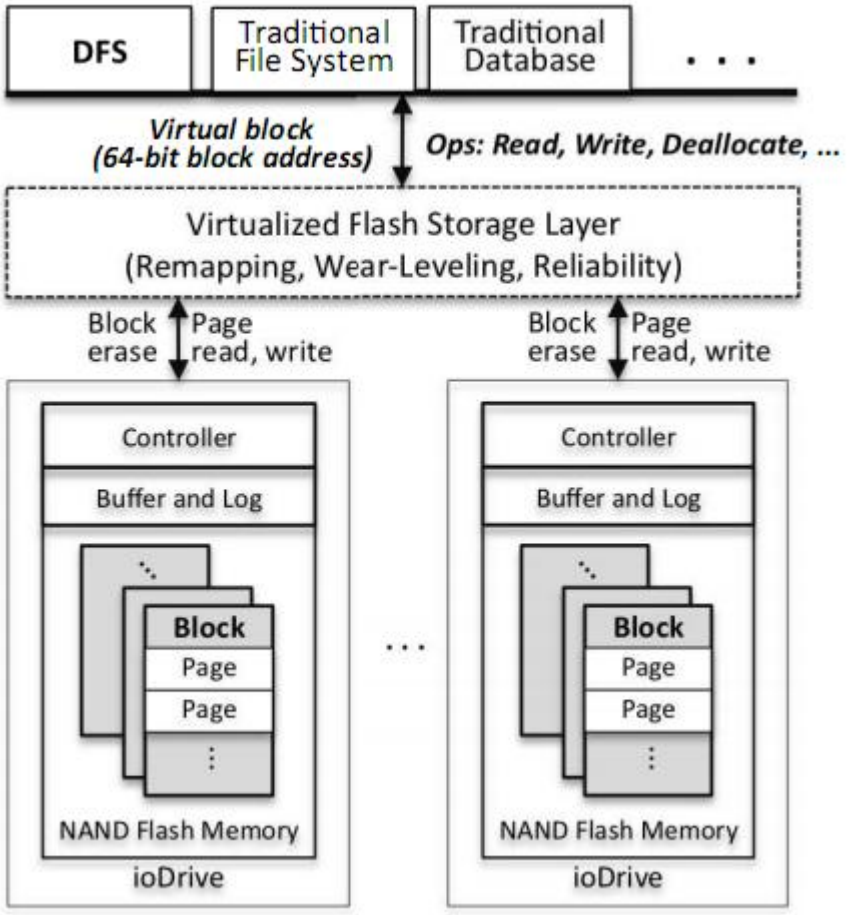
- 多采用sata接口，受限于sata的速度，无法发挥flash chip的全部性能。
- FTL隐藏了flash chip的特性，使得操作系统无法做专门的优化。
- 特别的linux内核把ssd当做一个块设备，使用块设备驱动模块进行使用，没有完全考虑到flash chip的block和page 的概念。

fusion-io

- IoMemory VSL

The virtualized storage abstraction layer provides a very large, virtualized block addressed space, which can greatly simplify the design of a file system while providing backward compatibility with the traditional block storage interface.

Instead of pushing the flash translation layer into disk controllers, this layer combines virtualization with intelligent translation and allocation strategies for hiding bulk erasure latencies and performing wear leveling.



DFS

- Vfs的存在使得文件系统和操作系统契合方面比较容易，只需要进行相关注册即可。
- 文件系统的功能就是向下管理块设备，向上提供文件视图，提供容灾和管理功能等。
- VSL提供了flash device的管理方案，所以DFS的实现借助于VSL省去了设备管理的复杂性，代码量小，稳定性好。
- 因为VSL的存在，大小文件逻辑上分离，每个文件分配固定大小的逻辑空间。
- 逻辑上分开的data path和control path。

性能评价

- 单片flash pkg内部执行命令流水线化，遵循onfi标准的nand芯片读写指标固定，对ssd性能没有影响。
- FTL算法对性能的影响很大，FTL算法很难做到通用，针对不同应用模型，diy空间很大。
- 顺序读写、随机读写性能，寿命、写放大等。

reference

- 从2010 IDF 看英特尔固态硬盘现状及策略
- DFS: A File System for Virtualized Flash Storage
- Design Tradeoffs for SSD Performance
- Algorithms and Data Structures for Flash Memories
- FTL Design Exploration in Reconfigurable High-Performance SSD for Server Applications
- DFTL: A Flash Translation Layer Employing Demand-based Selective Caching of Page-level Address Mappings
- A survey of Flash Translation Layer
- http://idke.ruc.edu.cn/people/dazhou/reference.htm
- http://www.linux-mtd.infradead.org/index.html
- http://www.fusionio.com/press/
- The SSD Relapse: Understanding and Choosing the Best SSD
- **SSD的写入放大 - Write amplification**