# 物联网数据存储与管理
# 概述

华宇

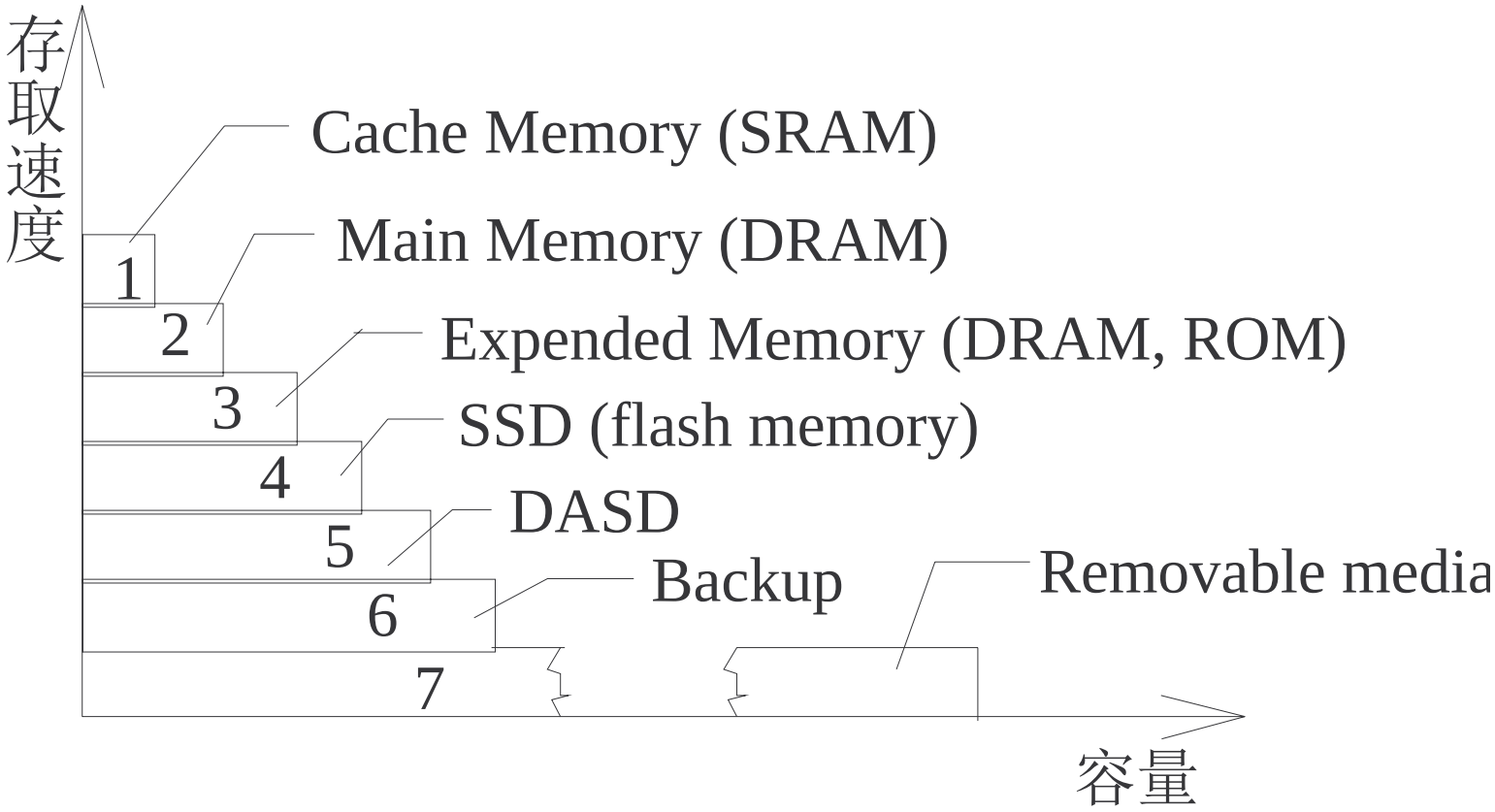
https://csyhua.github.io/

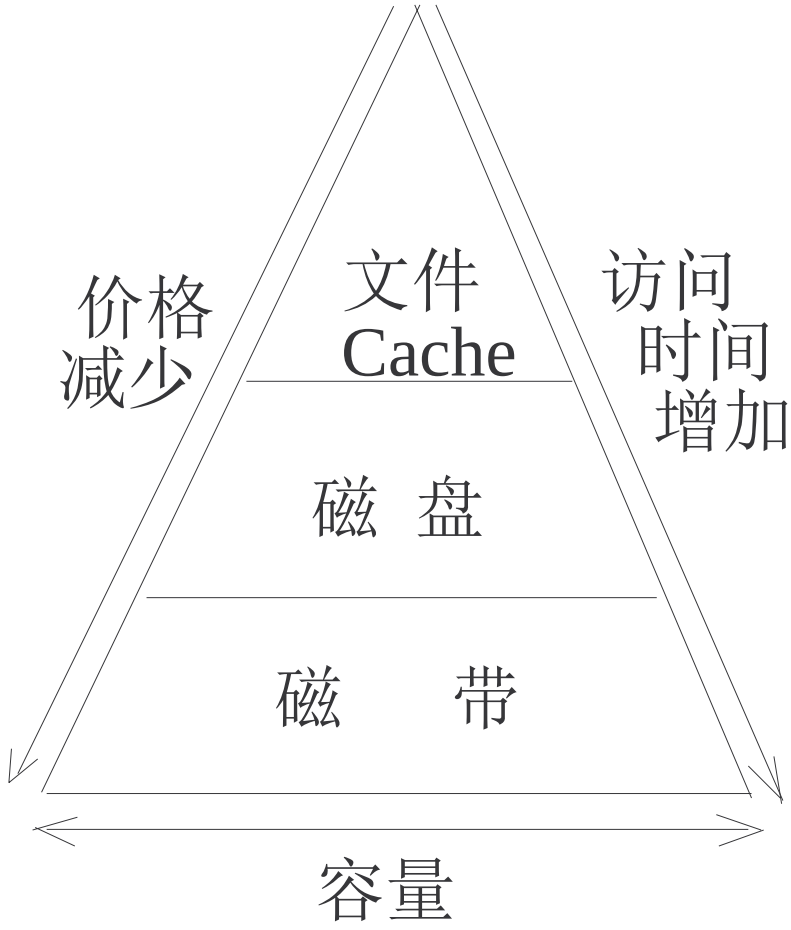
# 存储介质和设备

- Magnetic:
  - Hard drive
- Optical:
  - CD/DVD/Blue_Ray/etc.
- Solid State Semiconductor: fast growing!
  - Flash SSD
  - RAM SSD
- Tape: sequential accessed

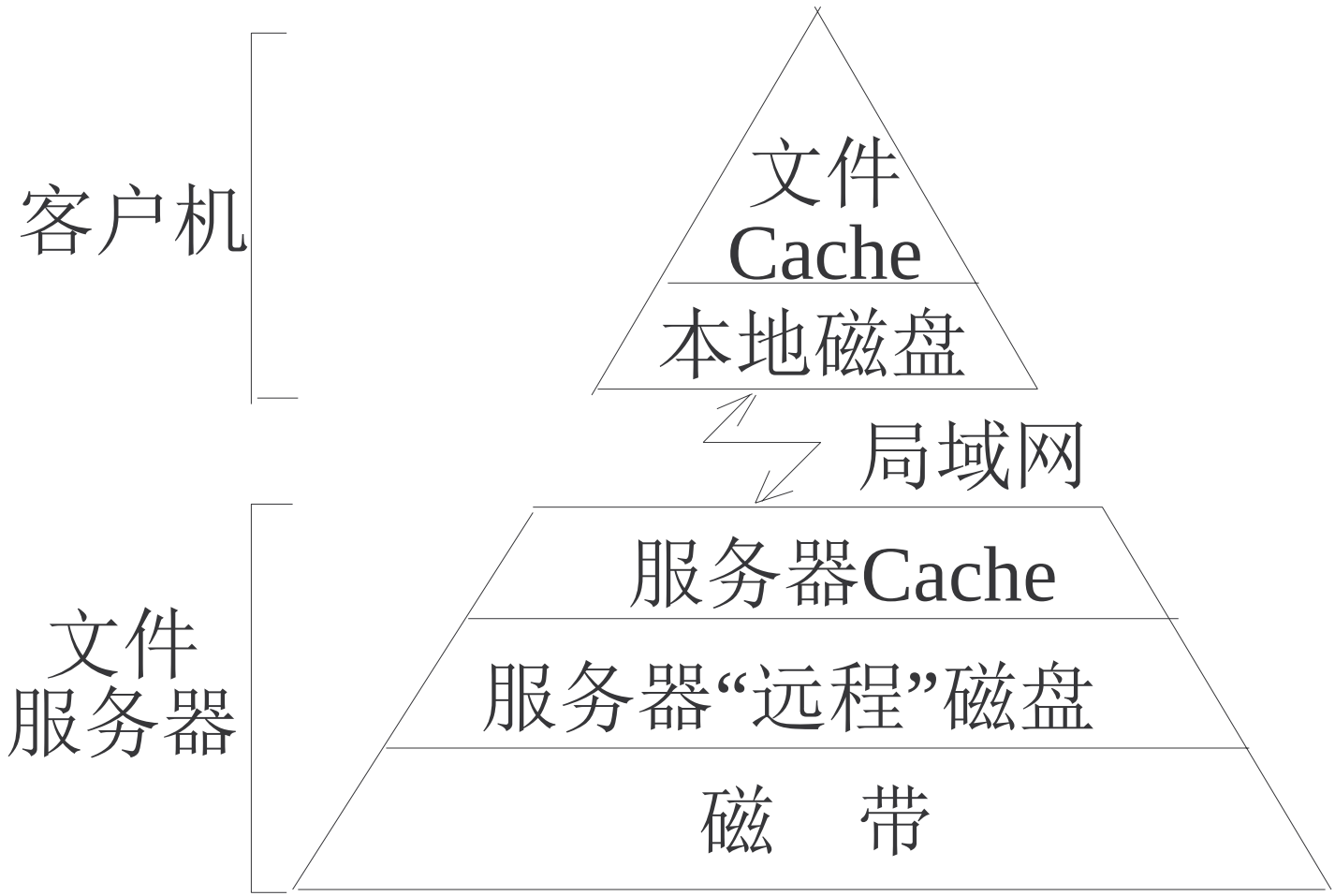
# 存储器分层结构

存储器设计的三个问题：容量、速度、价格

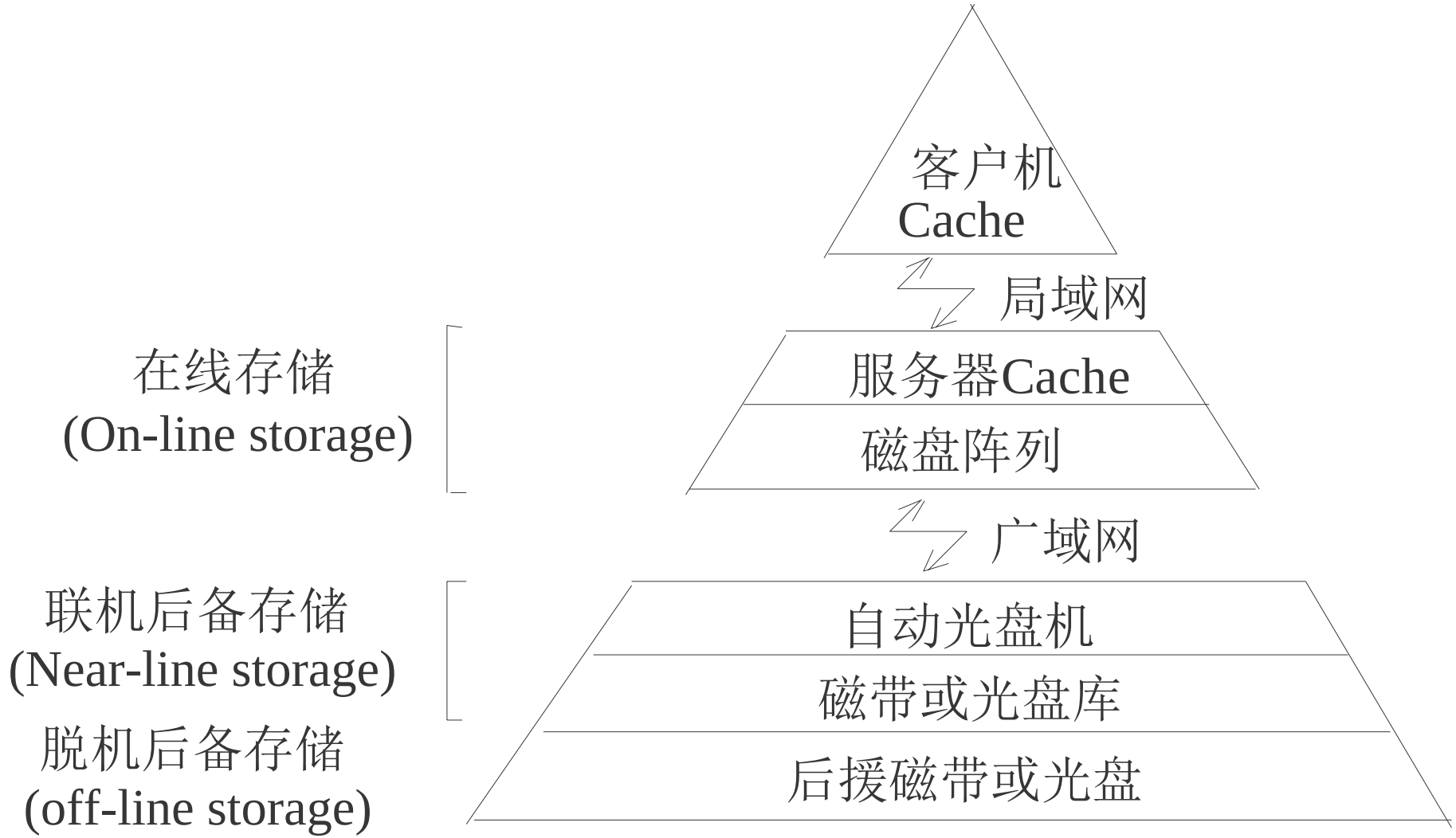




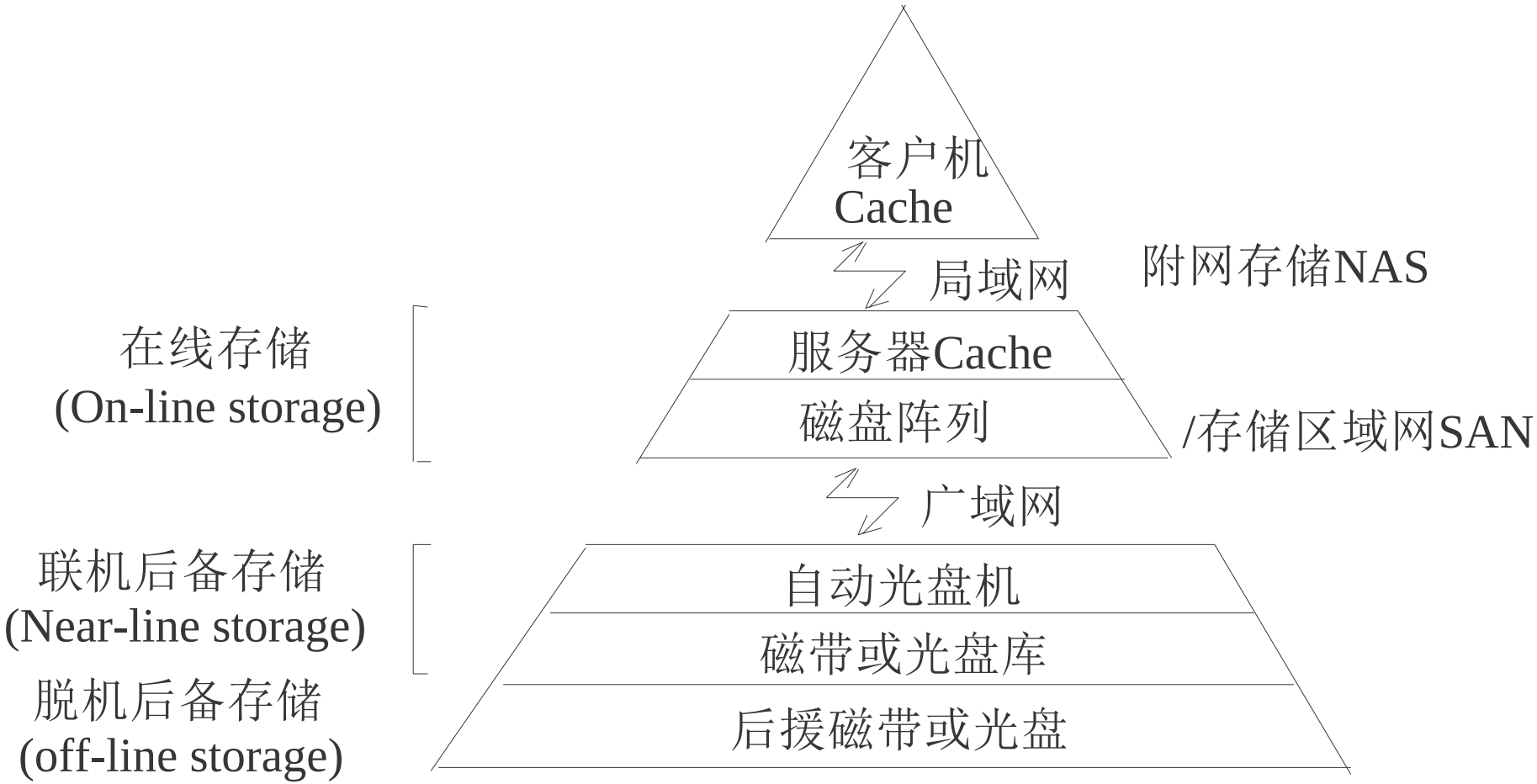

(a) 网络存储系统层次结构(1980年)




(b)　网络存储系统层次结构(1990年)




(c)　网络存储系统层次结构(1995年)




(d)　网络存储系统层次结构(当前)

## 光存储器

利用微小的激光束照射光记录媒体上，使被照射部位发生热效应或光效应，从而改变媒体的光学（或光磁）性质以记录信息的一类存储设备。读出时，媒体表面的状态转变为反射光强或偏振光的偏转角旋转，还原出记录的信息。

种类：

- 只读光盘存储器
- 只写一次读多次光盘存储器
- 可擦光盘存储器

# 发展趋势

存储领域中存在的“供不应求”状况。

需求:

Infinite Bandwidth

Infinite Storage

Infinite Processor Power

研究:

高速度：高速接口技术；

**Ultra3 SCSI、FC、Serial ATA**

超大容量：系统技术；

高可靠、高可用：容错技术。

安全、可信存储

包括多个层次：器件、设备、系统、服务

# Interfaces

- Internal
  - SCSI: wide SCSI, Ultra wide SCSI…
  - IDE ATA: Parallel ATA (PATA): PATA 66/100/133
  - Serial ATA (SATA): SATA 150, SATA II 300
- External
  - USB (1.0/2.0)
  - Firewire (400/800)
  - eSATA: external SATA

# 新型存储技术

- 半导体存储：**RAM, ROM, FLASH，PRAM, MRAM, RRAM,STTRAM，…**的高速发展
- 目前硬盘存储:

  **200GB; 2**万转/分；实验室水平：一道一**G**
- 光存储：**CD-ROM，DVD，MO**
- 近场光记录，全息存储
- 量子存储、生物存储
- **……**

# 磁盘的问题

- 磁盘由盘片，磁头，磁头臂及相应的电子器件组成
- 磁盘内存在两个马达，驱动盘片的马达和驱动磁头臂的马达
- 读写擦除数据的过程是：寻道（移动磁盘臂）→寻找扇区（转动盘片）
- 带来的问题是：小读，小写性能差（多次寻道）；能耗大（机械设备耗能）

# 非易失半导体存储器

- 在存储系统中的应用
  - 基于非易失半导体的固态硬盘
    - 高端个人电脑，笔记本电脑，移动设备
  - 在存储系统中作为主存和硬盘之间的中间层
    - 基于应用的局部性，把经常会访问到的数据存储在闪存中，从而改进存储系统的整体性能
  - 在存储系统中作为元数据存储器

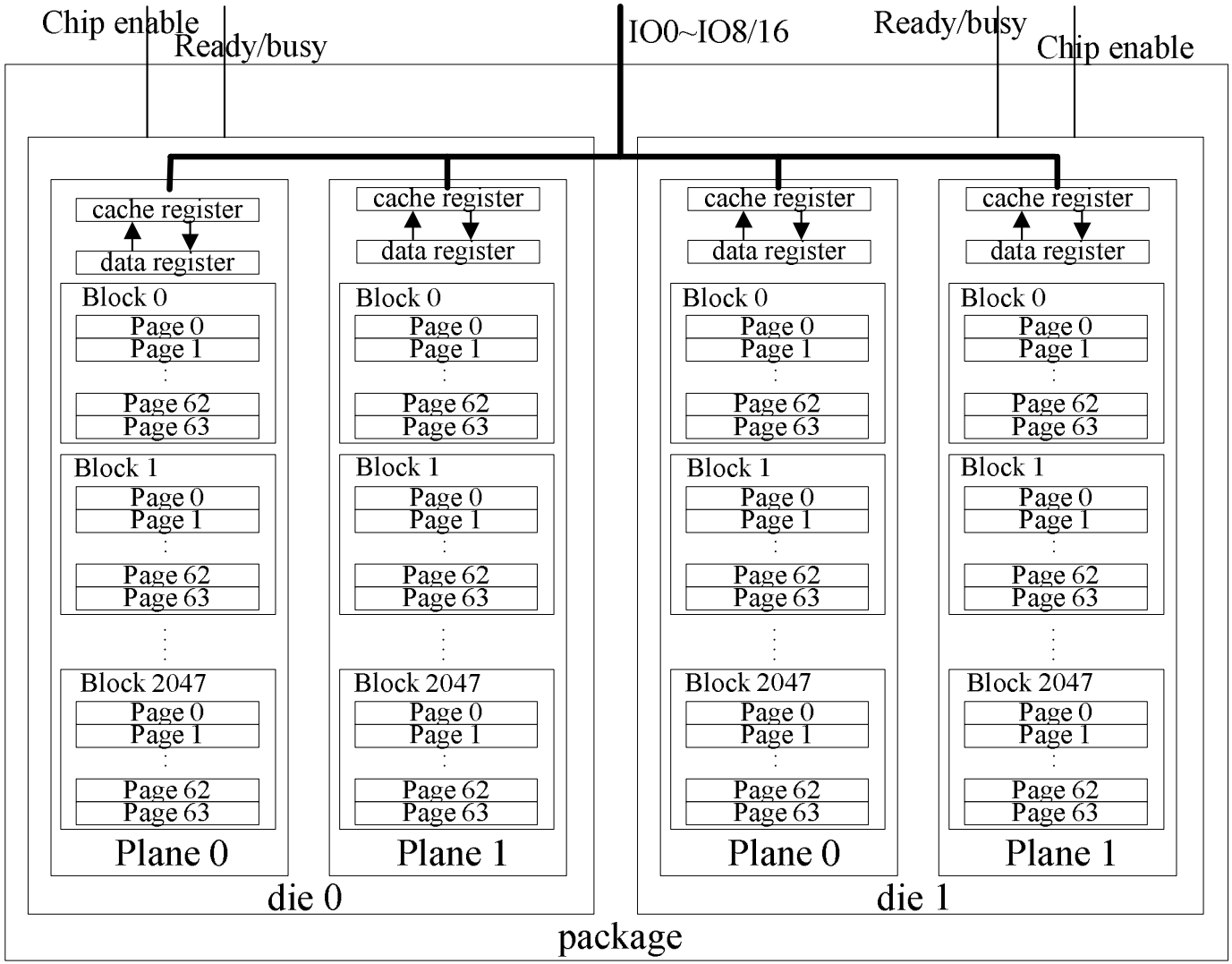
# 新型非易失存储器

1. Flash Memory – 闪存
2. Phase Change memory (PCM)--相变存储器
3. Memsistor
4. FeRAM
5. MRAM
6. ……

# 闪存

- 利用闪存作为存储介质，构建基于闪存的固态盘（flash-based solid state drive,SSD）
- 优点，小读请求很好，能耗低。
- 缺点，先擦后写，擦写次数有限制。

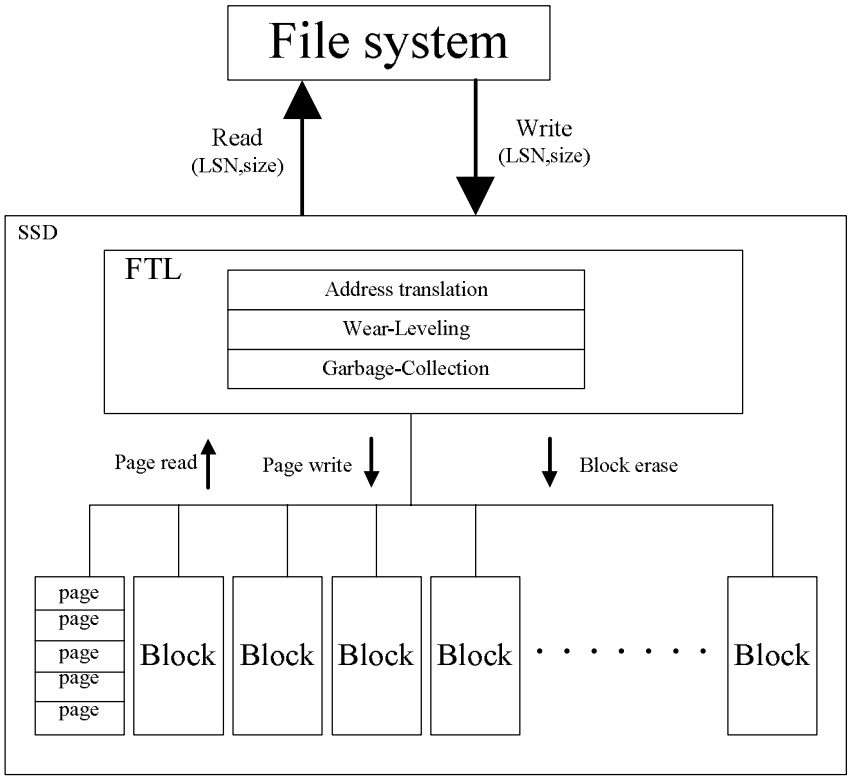
# 闪存的结构


Chip enable
Ready/busy
IO0~IO8/16
Ready/busy
Chip enable
cache register
data register
Block 0
Page 0
Page 1
Page 62
Page 63
Block 1
Block 2047
Plane 0
Plane 1
die 0
die 1
package

# 闪存，SSD最新研究进展（1）

- 与磁盘相同，SSD中存在一定的内存容量，这个内存可以被用作读写缓存（buffer）
- 在磁盘中RAM buffer的使用，较好的掩盖了磁盘的低效随机写操作；在SSD中，同样需要一个优秀的buffer算法来提高SSD的随机读写性能。

# 闪存，SSD最新研究进展（2）

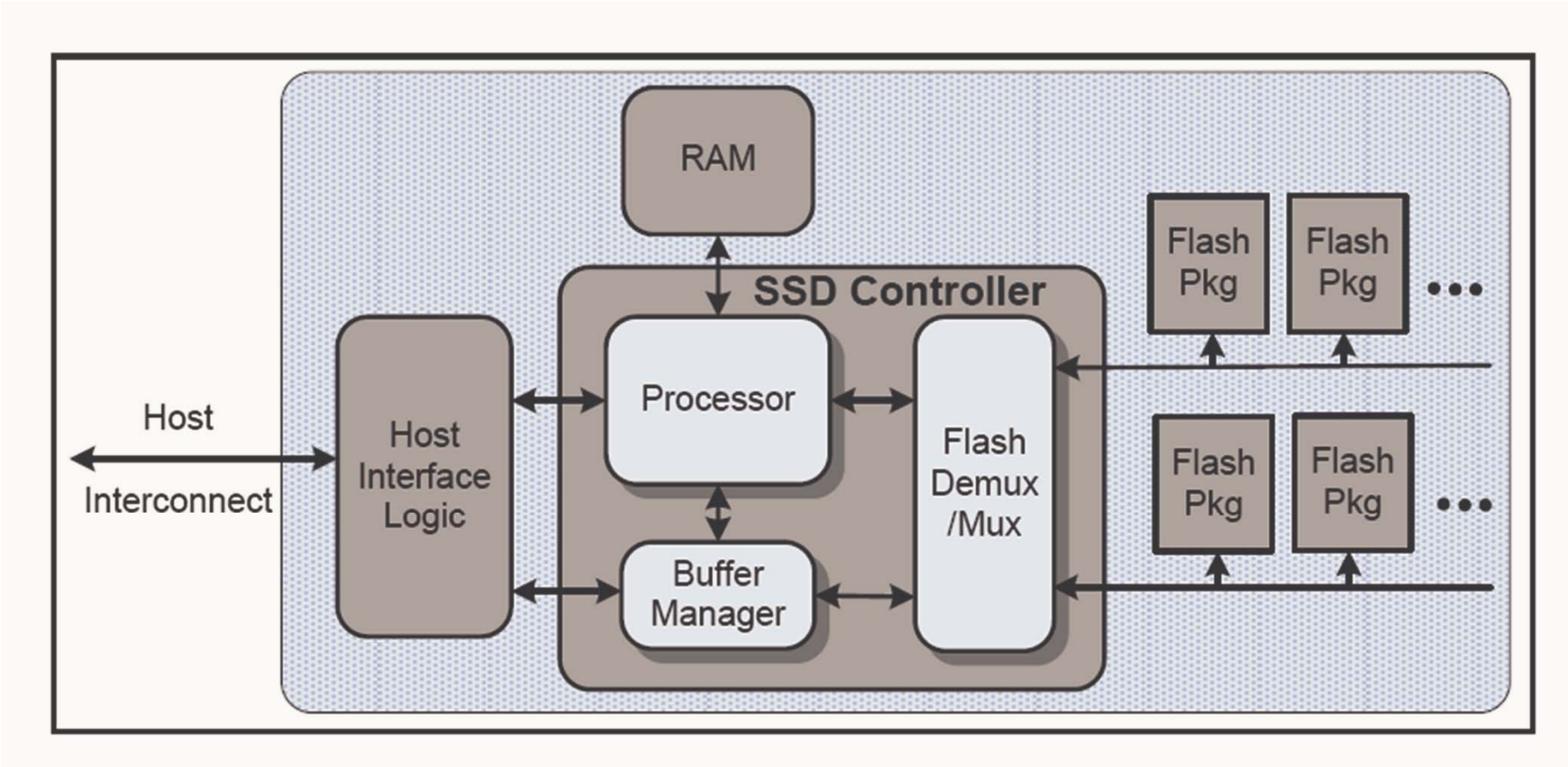



- 闪存本身有天然的缺陷：先擦后写，擦写次数有限制。为了提供一个理想的介质给上层文件系统，一个软件中间层（Flash Translation Layer）是必须的。

1. 地址转换 （Address Translation）
2. 损耗平衡 （Wear Leveling）
3. 垃圾回收 （Garbage Collection）

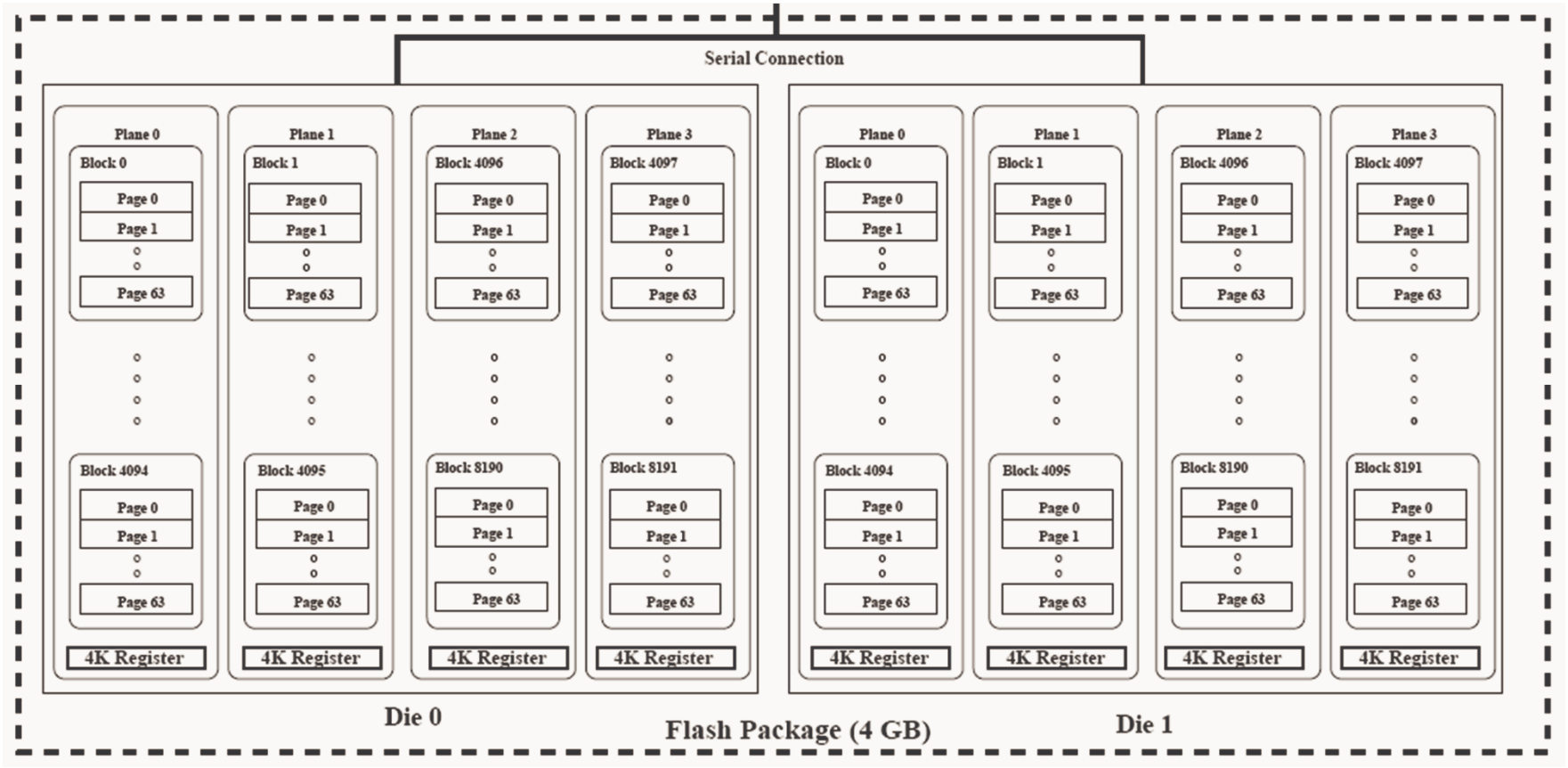
# Solid State Disks (SSDs)

- SSD Logic components



# Solid State Disks (SSDs)

- SSD core:  flash internals

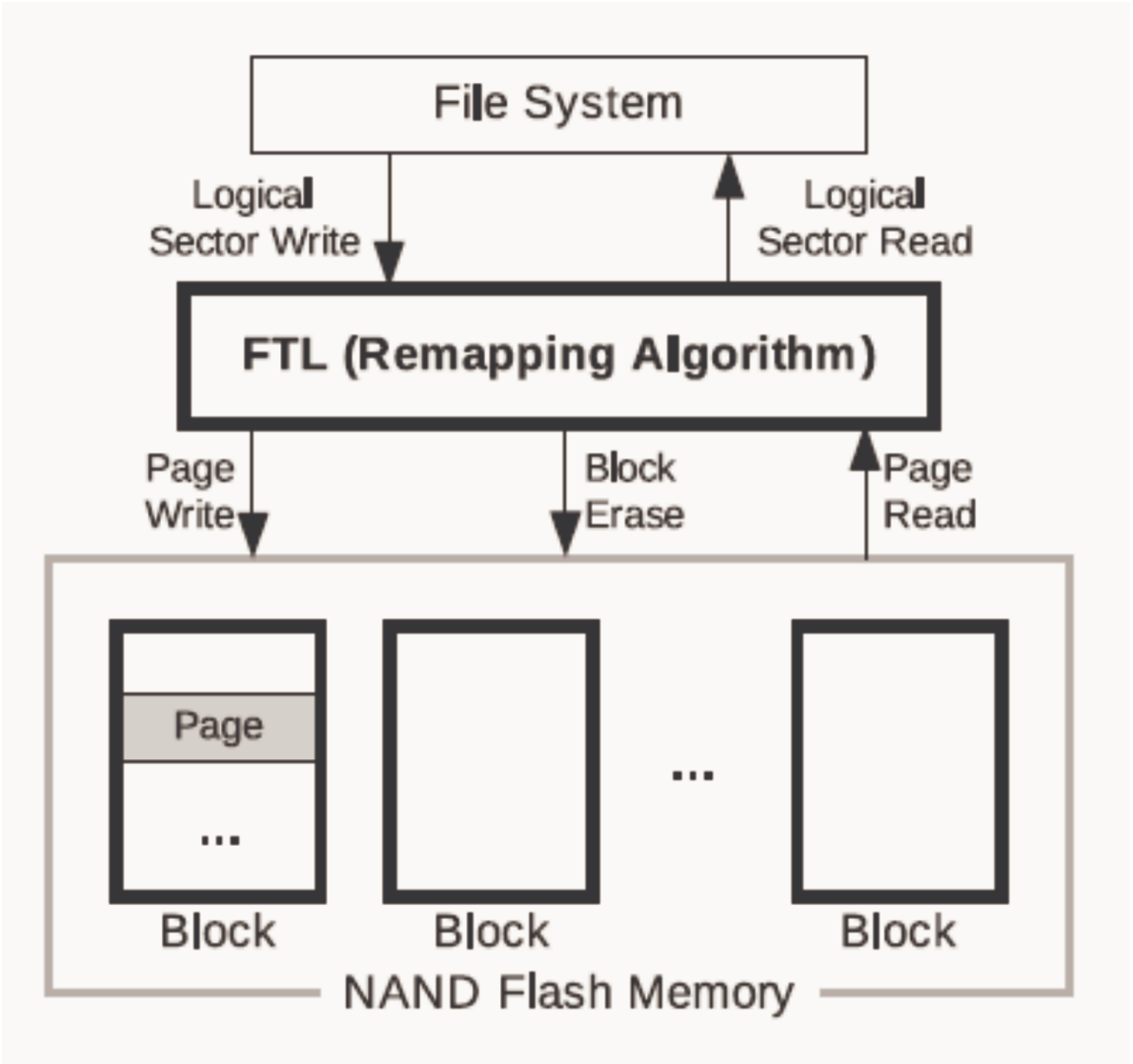


# Solid State Disks (SSDs): FTL

The FTL provides

- logical-to-physical address mapping
- power-off recovery
- wear-leveling.

# Solid State Disks (SSDs): FTL



# 地址转换（1）

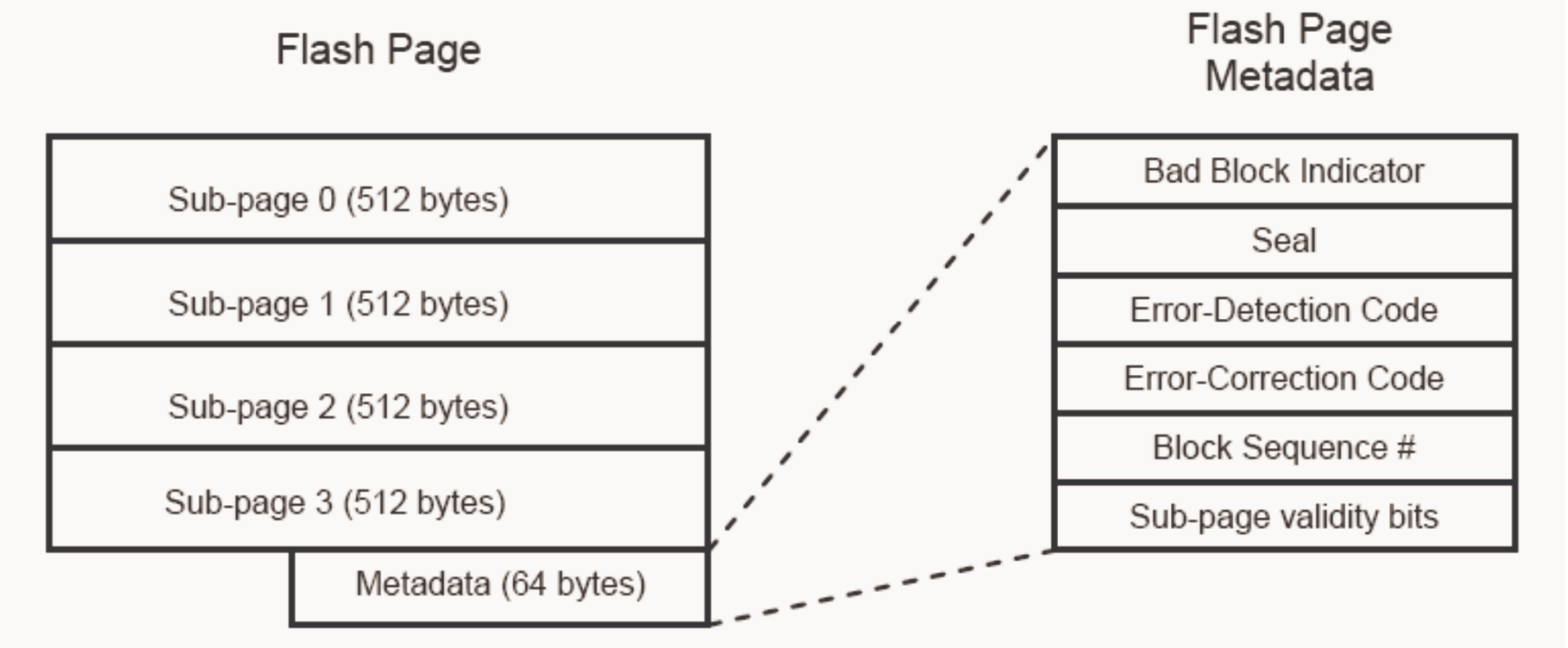
- 页映射（page mapping FTL）以页为映射单元，可将逻辑页映射到flash中的任何物理页。性能好，但是需要维持很大的映射表（假设每条映射关系需要4B，一个128GB的SSD的页映射表有128MB）
- 块映射 （block mapping FTL）以块为映射单元，将逻辑页映射到flash块中固定的物理页。映射表比较小（1MB），但是过多的擦除操作带来很大的性能损失。

# 地址转换（2）

- 混合映射（hybrid FTL）。将flash中的块分成两种类型：data block，log block。Data block采用块映射，log block采用页映射。混合映射兼顾了性能和RAM容量，在大多是SSD产品中采用。
- 最新提出的新的混合映射，包括FAST，LAST，superblock等，目的都是减少合并操作。

# Solid State Disks (SSDs)

## Flash Page Layout



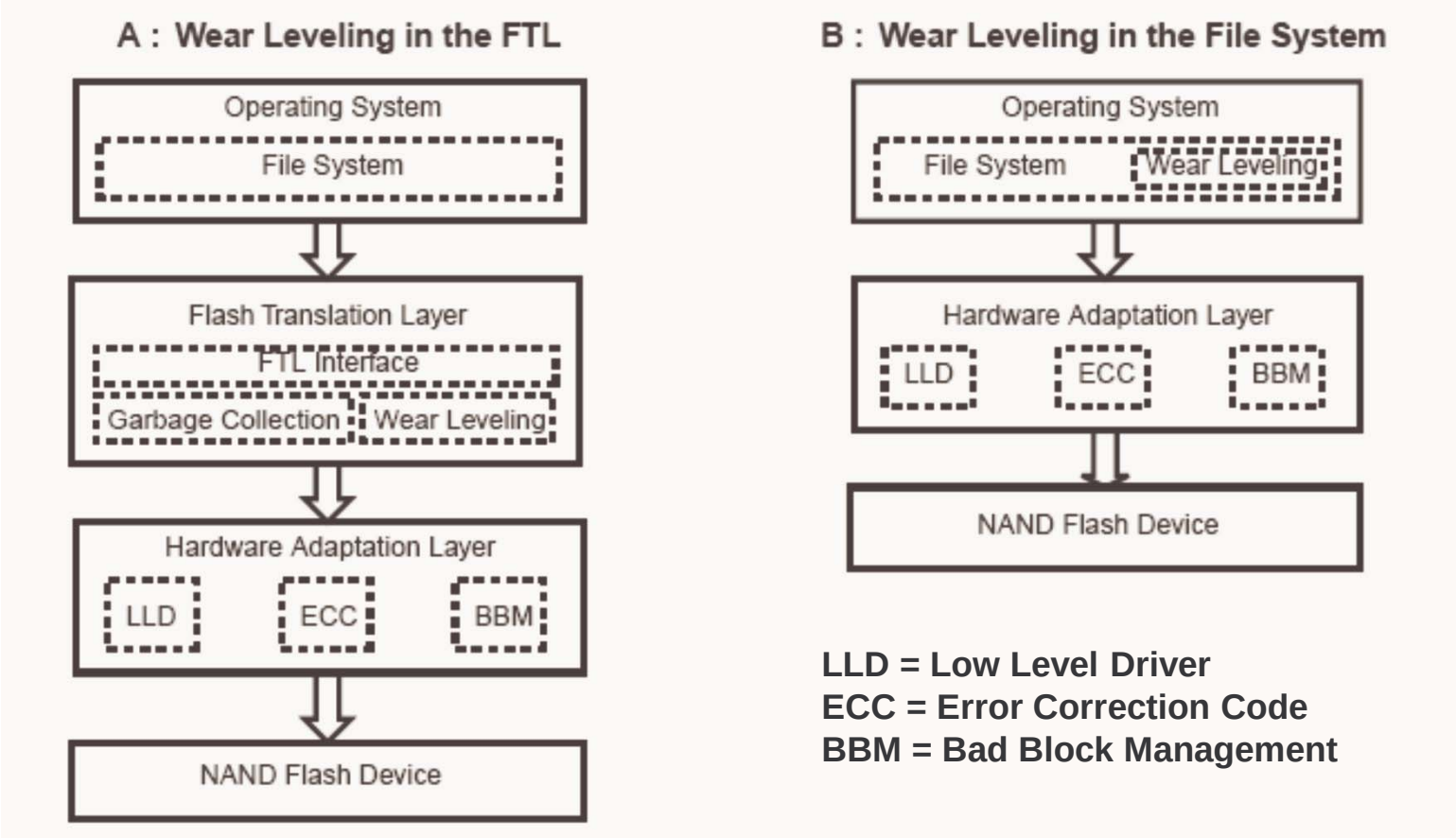
# Solid State Disks (SSDs)

Operational flash parameters

| | |
|---|---|
| Page Read to Register | 25$\mu$s |
| Page Program (Write) from Register | 200$\mu$s |
| Block Erase | 1.5ms |
| Serial Access to Register (Data bus) | 100$\mu$s |
| 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 |

# Solid State Disks (SSDs)

## *Wear Leveling* in NAND Flash Device



# An example of SSD

- 1 die = 4 planes
- 1 plane = 2048 blocks
- 1 block = 64 pages
- 1 page = 4KB
- Dies can operate independently
- Reading and programming is performed on a page basis, erasure can only be performed on a block basis.

# An example of SSD

- Read
  - 25µs from page to data register
  - 100µs transfer in the serial line
- Write
  - Page granularity
  - Sequentially with in a block
  - Block must be erased before writing
  - 200µs from register into flash cells

# An example of SSD

- Block erasure:
  - The erase state: 0xFF or 0x00
  - 1.5ms ( 25µs for reading a page)
  - Finite number of erase-write cycles
- Cleaning:
  - Erase out-of-date pages
  - Garbage collecting

# Solid State Disks (SSDs): Advantages

- Reliability in portable environments and no noise
  - No moving parts
- Faster start up
  - Does not need spin up
- Extremely low read latency
  - No seek time (25 us per page/4KB)
- Deterministic read performance
  - The performance does not depends on the location of data

# Solid State Disks (SSDs): Disadvantage

- Cost significantly more per unit capacity
- Limited write erase cycles (update must first erase the entire block of typically 64 pages)
  - 100000 writes for SLC (MLC is even fewer)
  - high endurance cells may have an 1-5 million
  - But some files still need more
  - Weaver leaving to spread writes all over the disk
- Slower write speeds because of the erase blocks are becoming larger and larger(1.5 ms per erase)
- For low capacity flash SSDs, low power consumption and heat production when in active use. High capacity SSDs may have significant higher power requirements

# PCM

- 相变存储器利用相变材料具有晶态，非晶态两种状态来保持数据。（晶态：低阻；非晶态：高阻）
- 相变存储器不需要擦除操作（in-place program），可以按字节进行访问，并且每位可写的次数是flash的10倍以上。
- HAT中将映射关系存放在PCM上，利用了PCM可以按字节访问的特写（映射关系只有4B）。

# 相变存储器

- 利用硫族化合物在晶态和非晶态的导电性差异存储数据
- 优点
  - 不需要擦除
  - 比闪存有更快的写性能和随机写性能
  - 耐久性
- 缺点
  - 难于控制相变的温度
  - 散热
  - 相变的时间比动态随机访问存储器充放电的时间长

# 混合式存储节点

- 随着闪存flash，相变存储PCM的出现，传统的存储节点将发生较大的改变。
- 在原有的存储节点中，文件系统下来的数据直接写到磁盘中，备份数据写到磁带中。
- 现在数据先写到基于flash的SSD，经过一段时间再写到磁盘，备份数据直接记录在磁盘上。

# 非易失存储器对未来存储系统的影响

- 闪存已经得到了大规模的应用，但闪存的性能以及可靠性都比目前的主存要差，因此闪存只能作为数据存储或作为主存和外存的中间层
- 相变存储器拥有比闪存更高的性能以及耐久性，因此相变存储器在将来很可能会取代闪存。但相变存储器的速度仍然无法于动态随机存储器相比，因此无法取代主存

# 非易失存储器对未来存储系统的影响(续)

- 磁阻存储器拥有其他存储器的大部分优点，并且有动态随机存储器相当的性能，因此被认为会取代当前的主存。但磁阻存储器目前的发展远落后于闪存和相变存储器

# 大规模存储系统

# 大规模存储系统面临的挑战

- 海量数据规模巨大且指数增长

  TB -> PB -> EB -> ZB -> YB

- 操作延迟增加，反应缓慢

  ms -> s -> min. -> hour

- 单一维度属性片面描述文件

  文件名，文件大小，创建时间，等等

- 传统树型结构的制约系统可扩展性

  访问瓶颈，迁移代价，查询效率

# 大规模存储系统现状分析

- 主要原因
  - 静态、不灵活的I/O接口
  - 大多采用线性搜索方法
  - 缺少语义分析
- 这样，对于Billion甚至Trillion级的文件系统进行管理就显得无能为力
- 通常，元数据操作占整个文件系统操作的50%以上

因此，我们以元数据管理为突破口，开展相关研究工作。

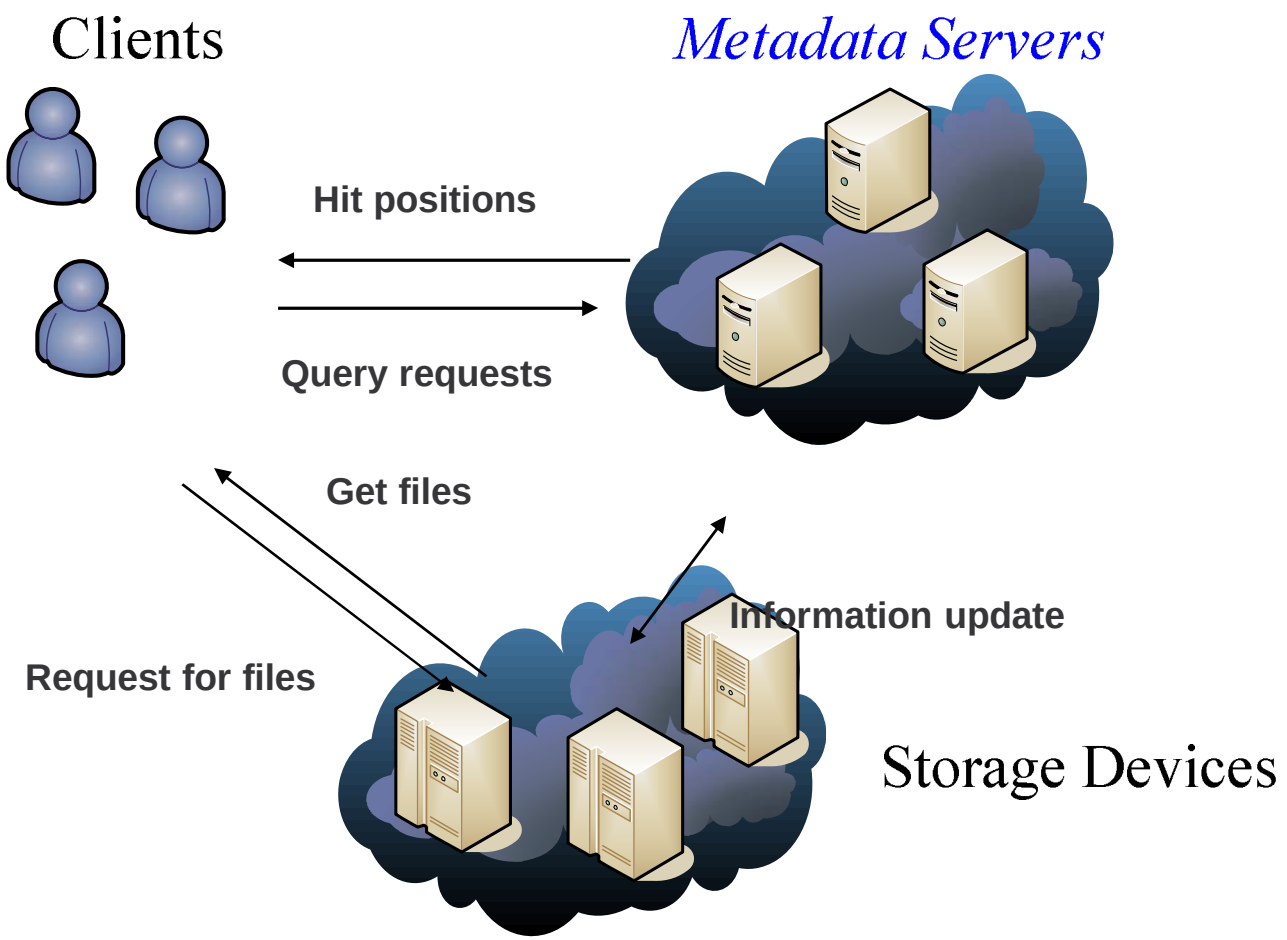
**提出的新方法：**

1. **Scalable and Adaptive Metadata Management in Ultra Large-scale File Systems (ICDCS-2008)**
2. **SmartStore: A New Metadata Organization Paradigm with Semantic-Awareness for Next-Generation File Systems (SC-2009)**

# Motivations

- Metadata management is critical in scaling the overall performance of large-scale data storage systems.
- Storage demands increase exponentially in recent years, exceeding *Petabytes* already and reaching *Exabytes* soon.
- Metadata transactions account for over 50% of all file system operations.

# A simple view



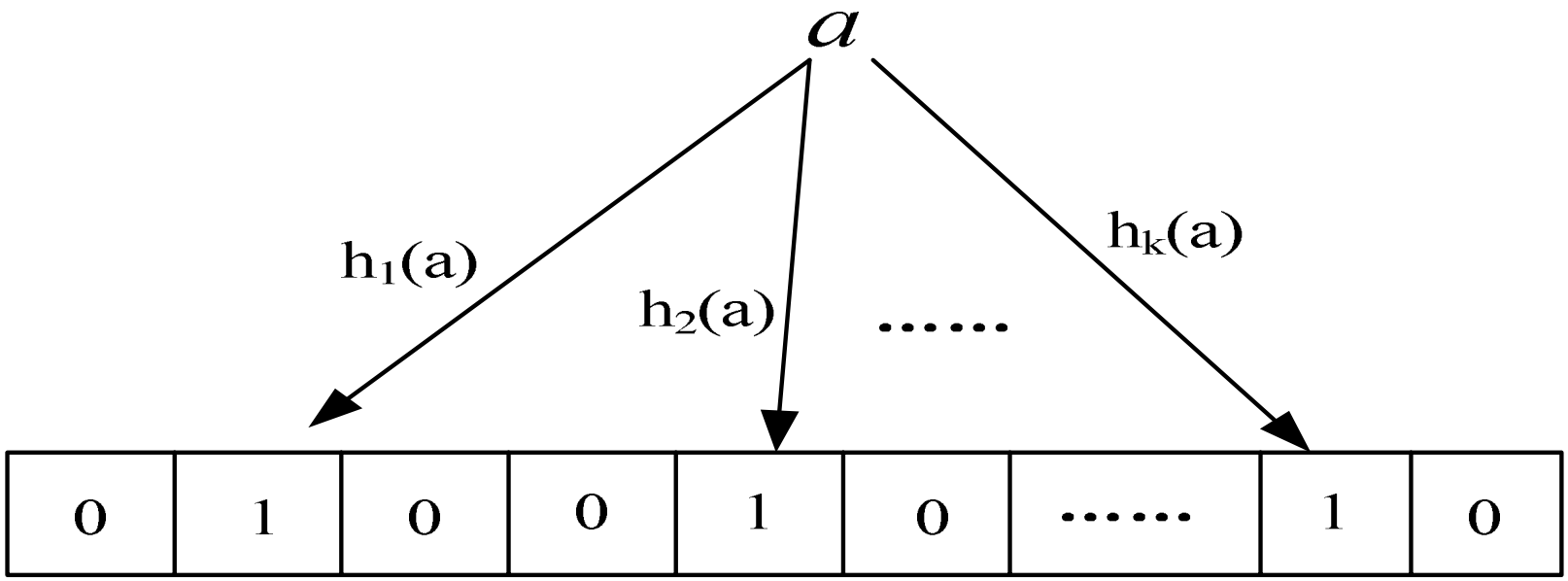
# Backgrounds and assumption

- An MDS where a file's metadata resides is called the *home MDS* of this file.
- Each metadata server constructs a Bloom filter to represent all files whose metadata are stored locally and then replicates this Bloom filter to all other MDSs.
- The Bloom filter array returns a *hit* when exactly one filter gives a positive response.
- A *miss* takes place when zero hit or multiple hits are found in the array.

# Bloom filter

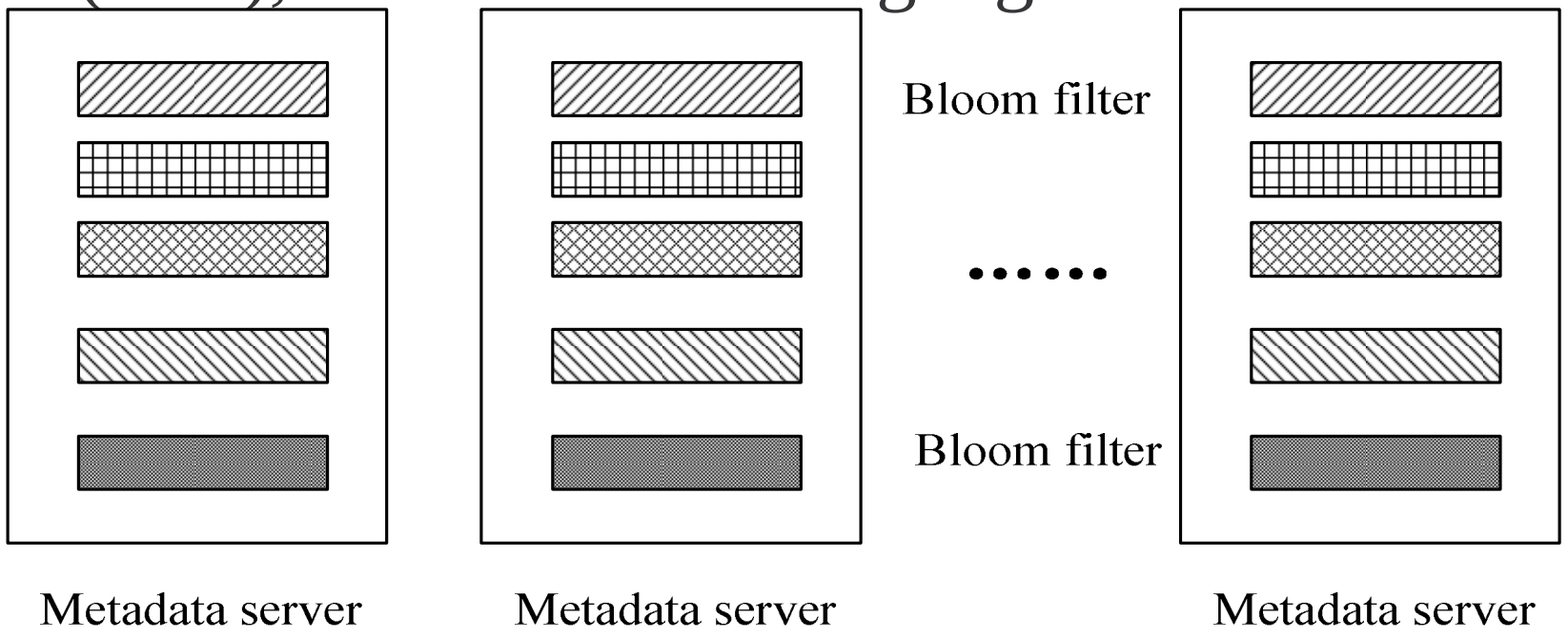
- Bloom filter uses an array of *m* bits (initially all set to 0) to describe a set $S=\{x_1,x_2,\ldots,x_n\}$ by *k* independent and uniform hash functions $h_1,h_2,\ldots,h_k$ with range $\{1,\ldots m\}$.
- For each item *x*, the bits $h_i(x)$ in the array are set to 1.
- A bit may be set multiple times, but only the first change has effect;

# Bloom filter

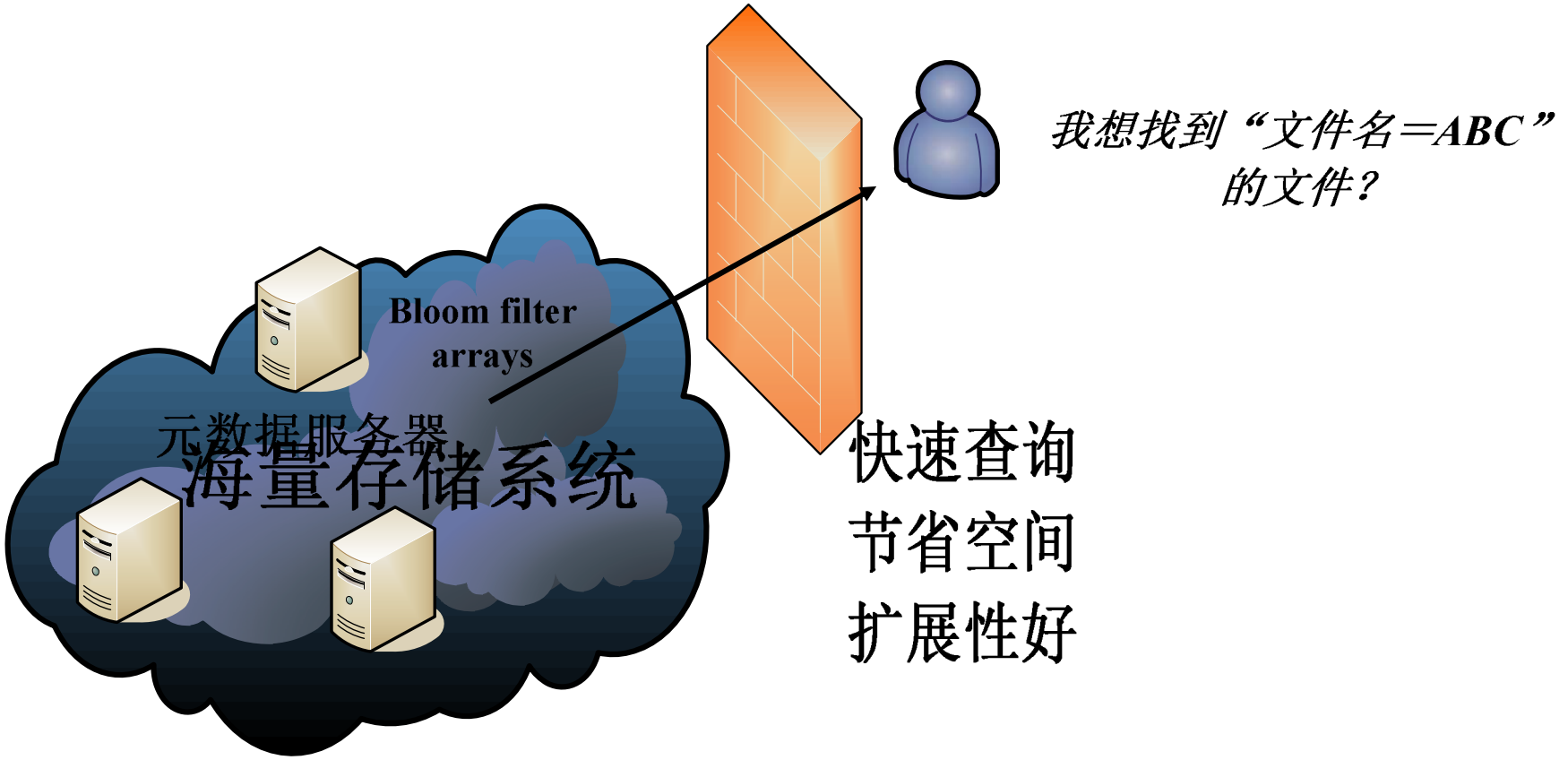


# BFA

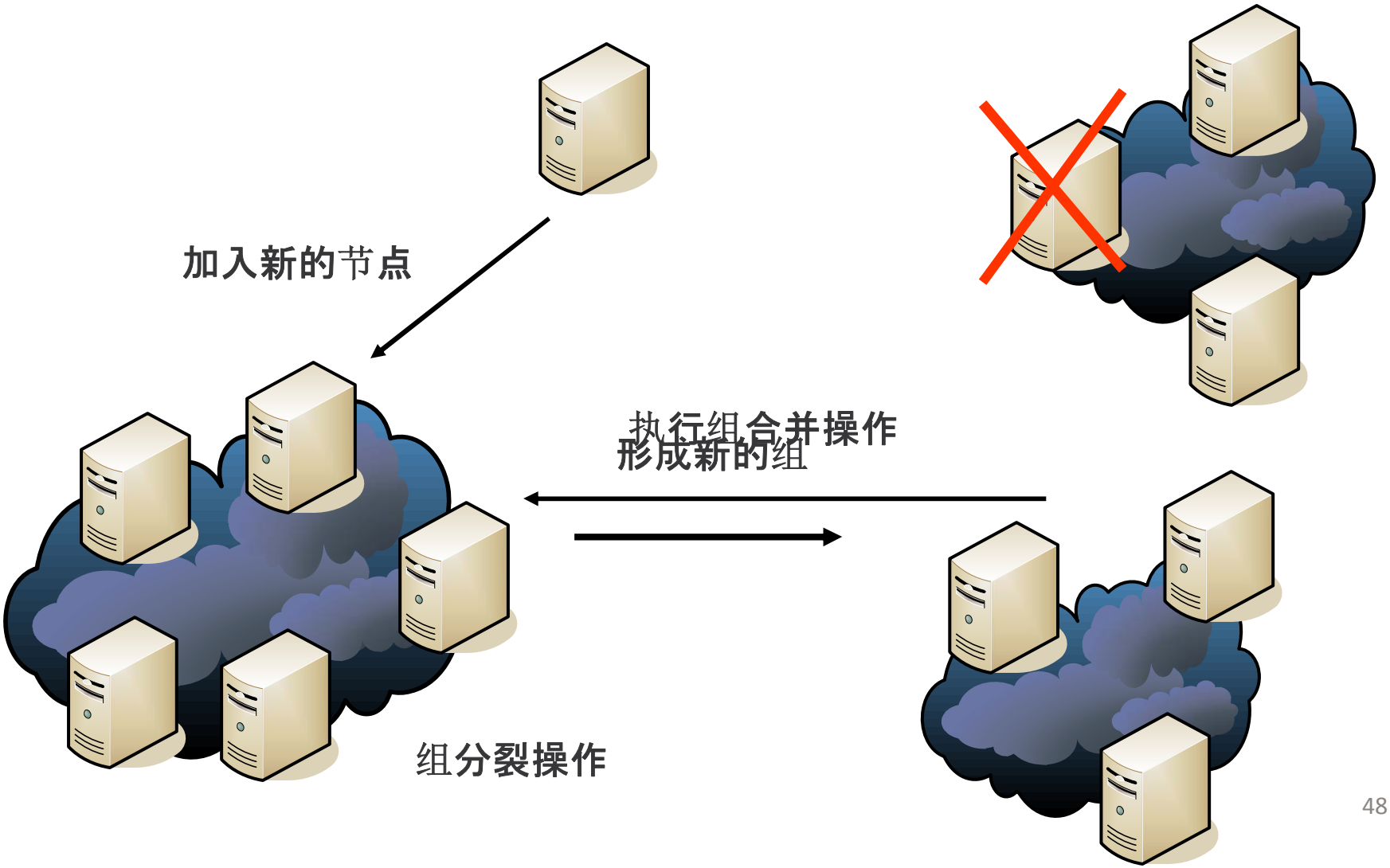
- The naïve solution is Bloom Filter Array (*BFA*), each MDS serving a global mirror.



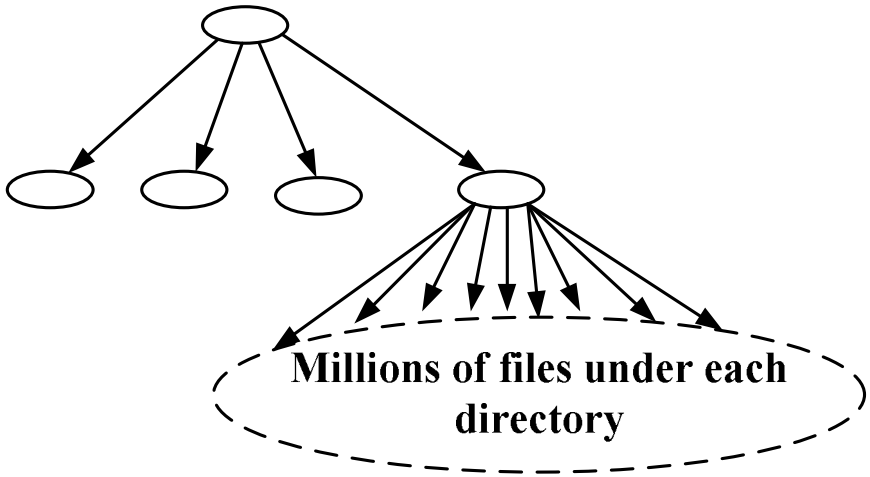
# G-HBA：提供快速点查询服务



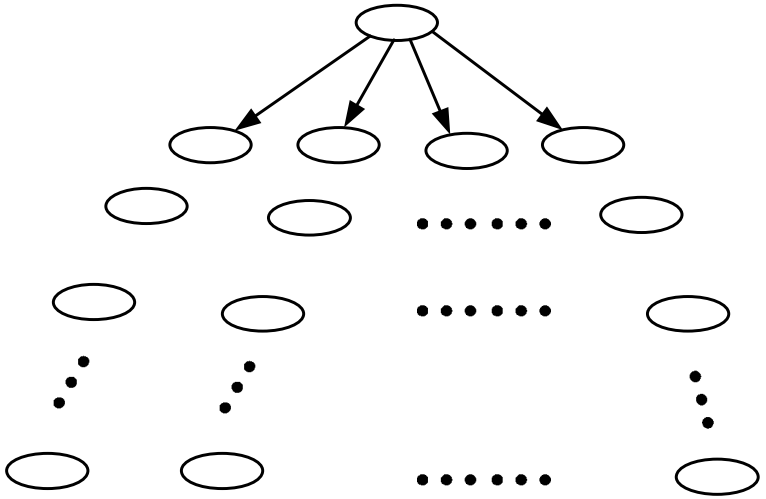
# G-HBA：提供自适应的存储查询服务



# 传统目录树组织模式的问题




“胖”树



“高”树



49

# Motivations

- Some Facts:

➢*Storage capacity -> Exabyte (or even larger);*

➢*Amounts of Files -> Billions*

➢*Metadata-based transactions -> over 50%*

- Ideal scenarios:

➢Obtain interested knowledge from data ocean;

➢Carry out quick query for *high*-dimensional data;

➢Users equipped with “*data map*”

# Motivations (con.)

- Inefficiency of current file systems:

➢ Strict hierarchical directory-tree based metadata management;

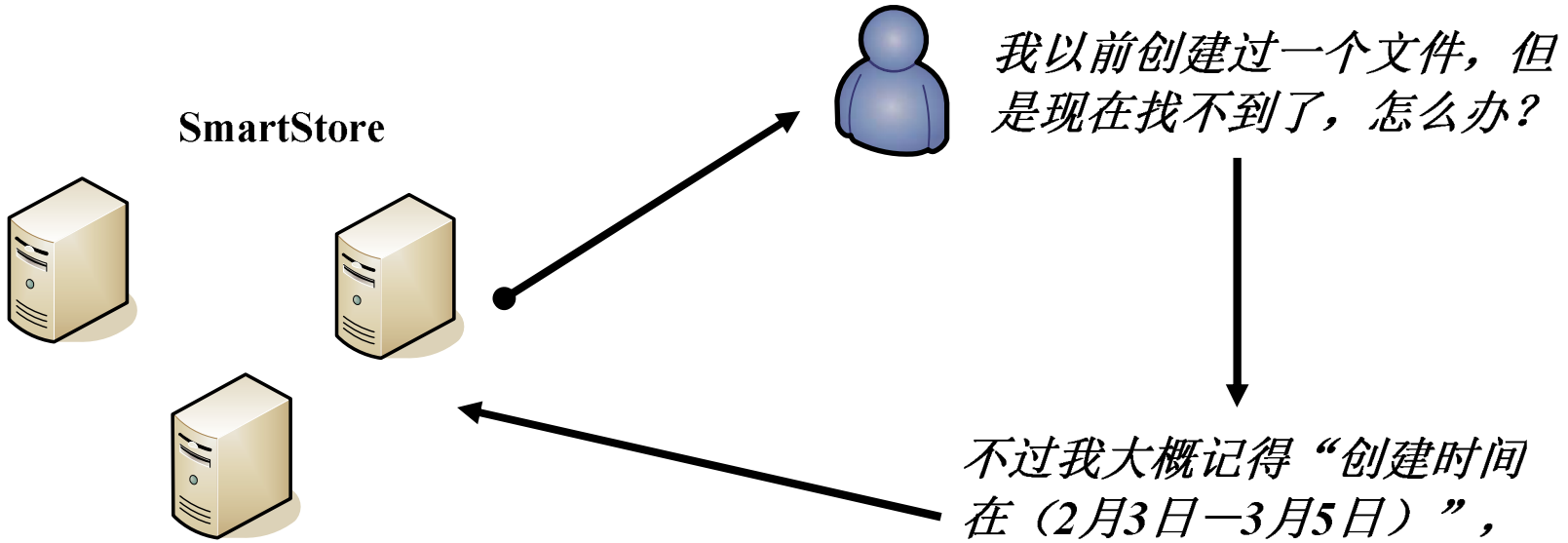
➢ Static and inflexible I/O interfaces;

➢ Linearly brute-force searching;

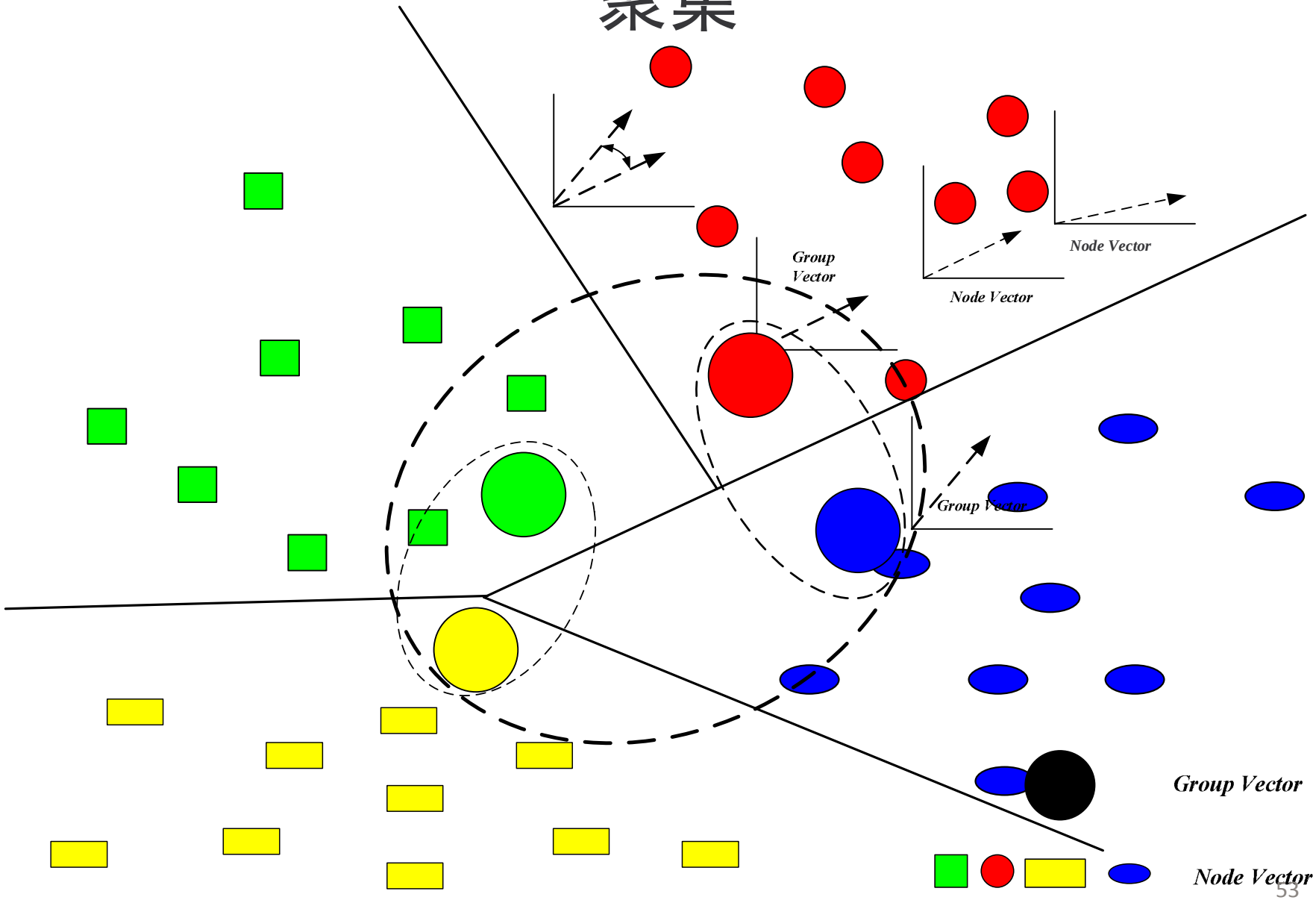
- What we do: *Allow users to locate target files in a large-scale storage system:*

❑ Desirable interfaces are *range query* and *top-k query, i.e., complex queries;*

# 支持复杂查询服务


SmartStore
我以前创建过一个文件，但是现在找不到了，怎么办？
不过我大概记得“创建时间在（2月3日—3月5日）”，文件大小（2KB—200KB）
通过语义信息聚集相关文件提供复杂查询服务

# 实现方法：通过语义分析工具进行分组聚集



# 未来元数据组织和系统管理的可能研究方向预测

- 海量元数据的组织和管理
  - 面向语义的数据组织
  - 基于数据内在关联关系的分析
  - 支持多种查询服务的方法
  - 基于用户访问模式的预测和分析
- 目标：
  - 面向用户提供可靠、高效的存储服务

# 未来元数据组织和系统管理的可能研究方向预测

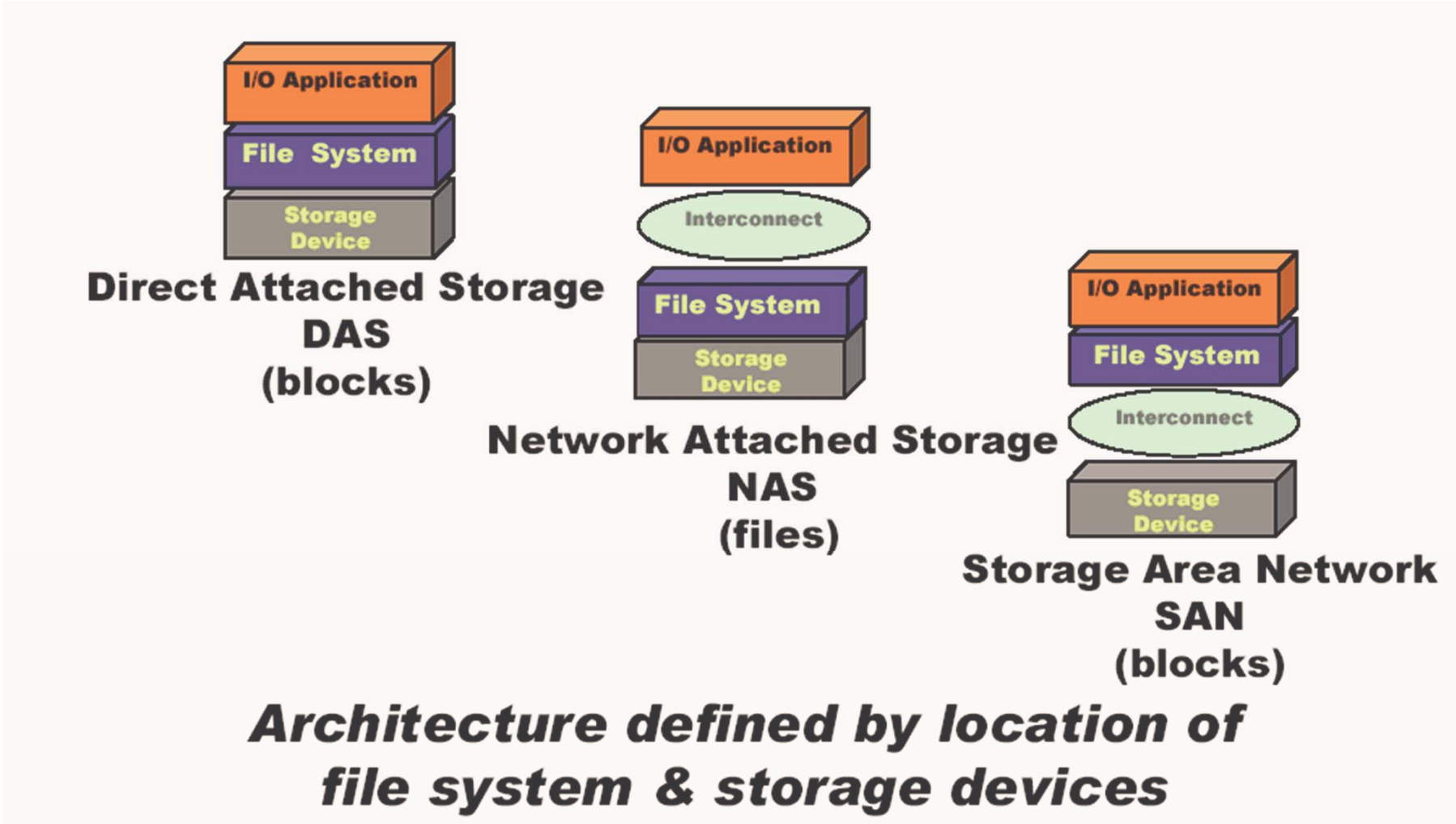
- 存储系统管理方面：

- 面向高复杂存储系统的自管理机制
- 提高系统智能性的自识别、自处理、自反馈、自调整的方法
- 面向应用的智能管理方法来提高存储系统的域名管理，系统恢复，健壮性等
- 技术点：缓存管理，预取技术，可靠性等

# 存储服务

- 云计算
- 云存储
- 云备份

# Storage Architectures

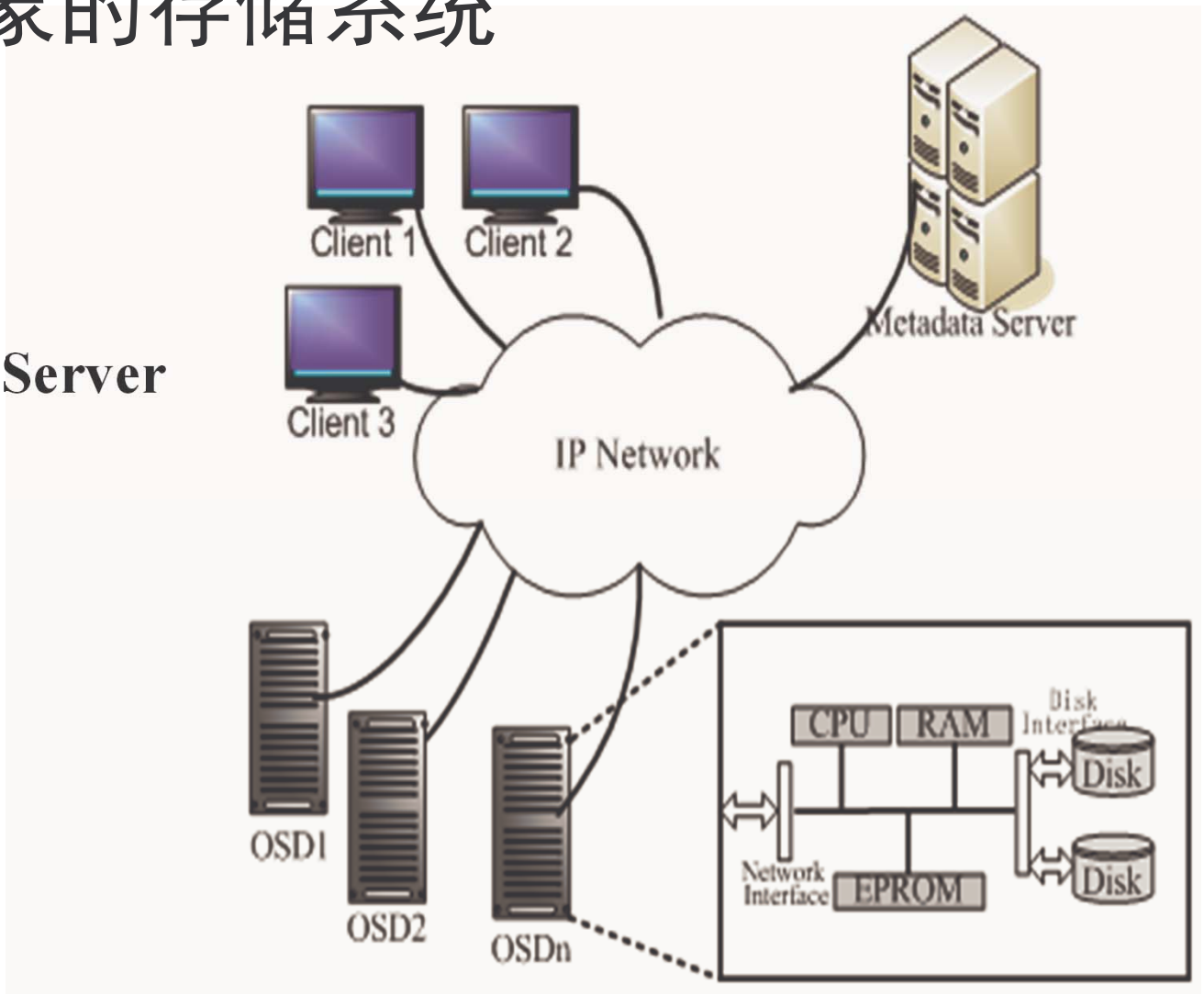


# 存储系统

- 直接附加存储（**Direct Attached Storage，DAS**）
- 附网存储（**Network-attached Storage，NAS**）
- 存储局域网（**Storage Area Network，SAN**）

# 基于对象的存储系统

- 组成:
  - **Metadata Server (MDS)**
  - **clients**
  - **OSDs**



# Hadoop Distributed File System (HDFS)

- Open source Map-Reduce
- HDFS is inspired by Google File System
- High fault-tolerant and designed to deploy on lower-cost hardware
- High throughput access
- Suitable for large data set
- Implemented by using Java, thus supporting multi-platforms.

# Assumptions and goals

- Hardware Failure
- Stream Data Access
- Large Data Sets
- Simple Coherency Model
- “Moving Computation is Cheaper than Moving Data”
- Portability Across Heterogeneous Hardware and Software Platforms

# NameNode and DataNode

- Master/Slave – NameNode / DataNode
- NameNode
  - Single NameNode
  - File namespace operations
  - Mapping block to DataNode
- DataNode
  - Serving reading/writing
  - Block creating/deleting/replication

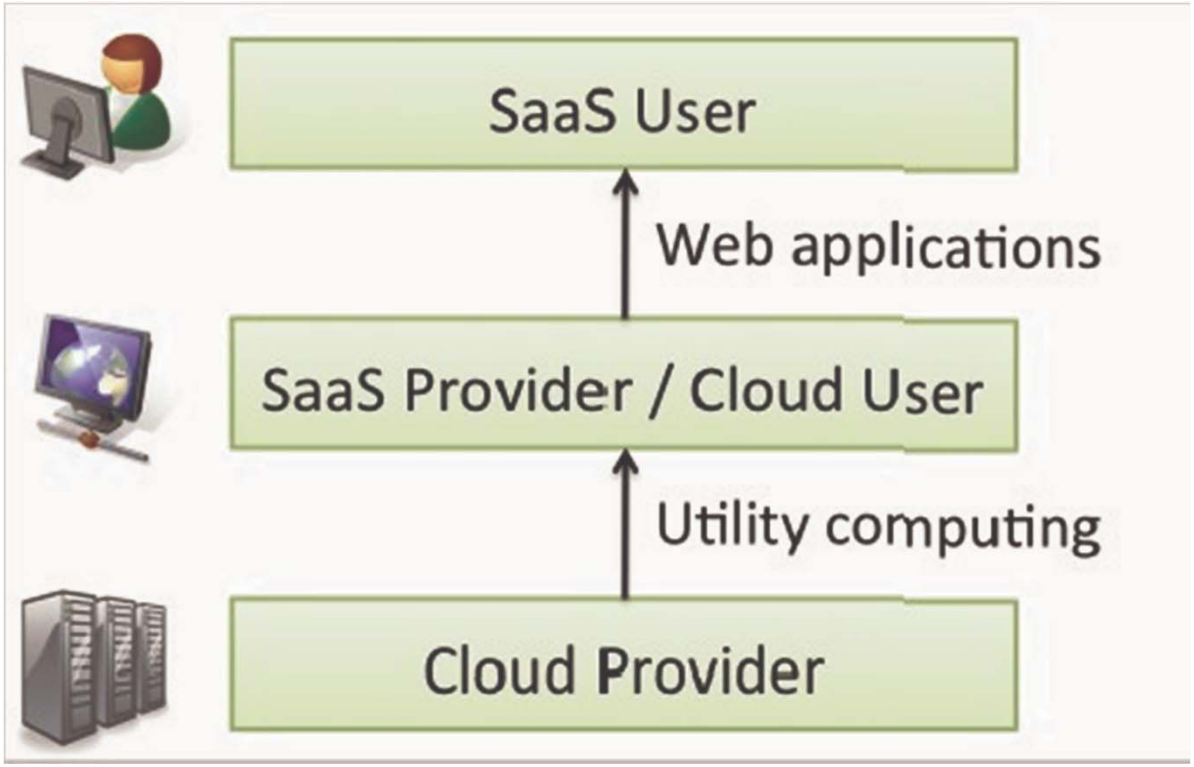
# 云计算 （1）

- 何为云计算（ Cloud Computing ）？

  云计算包括在Internet上发布的应用服务以及数据中心提供这些服务的软硬件。

  - Cloud：数据中心的软件和硬件。
  - SaaS：发布的应用服务。
  - Utility Computing：以 “pay-as-you-go”的形式出售的公共服务，如Amazon Web Services。
  - 云计算：SaaS 和 Utility Computing的总和。

# 云计算 （2）




其中，Cloud Provider 也可以是 SaaS Provider

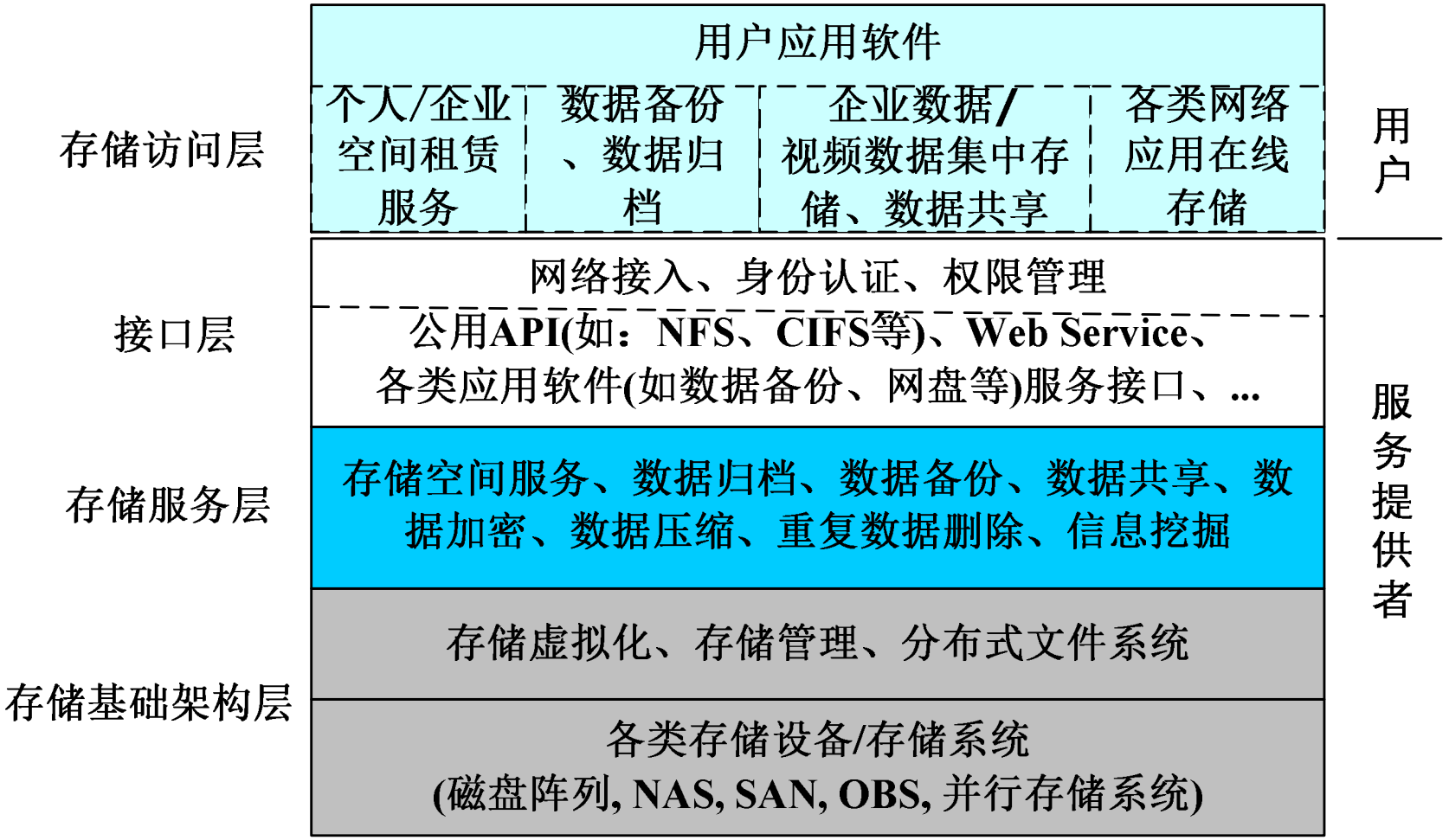
# 云计算 （3）

- 利用虚拟化的计算资源，存储资源，网络资源对外提供服务。
- 云计算在如下三个方面很新颖
  - 无限可用的资源
  - 用户（Cloud users）不需要预先配置资源
  - 资源的使用可以以时间为单位来付费

# 云存储

- 云存储（Cloud Storage）处于云计算范畴下。
- 云存储是一种存储服务，是云计算中Utility Computing的一个子集。
- 云存储服务的提供者（Cloud Provider）负责存储空间的可扩展性，存储的数据的可靠性，可用性等等。

# 云存储结构


用户应用软件
个人/企业空间租赁服务
数据备份、数据归档
企业数据/视频数据集中存储、数据共享
各类网络应用在线存储
网络接入、身份认证、权限管理
公用API(如：NFS、CIFS等)、Web Service、
各类应用软件(如数据备份、网盘等)服务接口、...
存储空间服务、数据归档、数据备份、数据共享、数据加密、数据压缩、重复数据删除、信息挖掘
存储虚拟化、存储管理、分布式文件系统
各类存储设备/存储系统
(磁盘阵列, NAS, SAN, OBS, 并行存储系统)
存储访问层
接口层
存储服务层
存储基础架构层
用户
服务提供者

# 云备份-概念 （1）

- 云备份（Cloud data backup）是云计算平台下一个非常典型的应用服务。
- 云备份将备份以服务的形式发布，用户将数据备份到云备份提供方。
- 云备份有两种实现方法
  - Backup SaaS。其中Cloud provider 和 SaaS provider 均为备份厂商，为客户提供备份服务。
  - 结合第三方的Cloud storage service providers。其中Cloud provider即Cloud storage service provider，而SaaS provider为备份厂商。备份厂商通过云存储服务提供者提供的API，将用户的备份数据存储在云存储服务提供方。

# 云备份 - 用户群

- 云备份的用户群：云备份很受个人用户和中小型企业（SMBs）的欢迎。
  - 个人用户。云备份可以将数据远程备份到云端，弥补了个人用户本地备份的不足。另外，个人用户可以随时随地的通过网络使用云备份服务，非常适合经常出差的办公者。
  - 中小型企业。中小型企业主要关注其商业的运转为企业带来的利益。中小型业规模较小，IT员工少，IT预算紧缺。而一套完整的备份系统很耗资源，包括硬件资源和软件资源，同时还需要专业的管理人员进行维护。中小型企业使用云备份，可以大大减少他们的负担，节约资源，使他们的精力能够集中到其商业作业中。

# 云备份 – 优劣势 （1）

- 云备份的优势（以下是部分优势）
  - 集中了现有的很多种技术，比如基于磁盘的备份，压缩，加密，重复数据删除，服务器的虚拟化，存储虚拟化，等等。
  - 异地备份。
  - 24 × 7 小时的备份/恢复服务。
  - 小数据集的恢复速度很快
  - 可扩展性很强
  - 经济实惠

# 云备份 - 数据传输瓶颈 （2）

- 解决数据传输瓶颈的方法有两种
  - 广域网数据服务（WDS, wide area data services）． WDS 通过重复数据删除和压缩的方式，来减少数据的传输。WDS 与应用无关。典型的产品为 riverbed。
  - 重复数据删除（data de-duplication ）． 针对于不同应用的重复数据删除的实现方式不同。
  - 以上两种方式都是通过减少数据的传输量，来减少低带宽，低延迟的广域网对数据传输带来的影响，消除数据传输瓶颈。

# 云备份-重复数据删除（1）

- 重复数据删除是一个无损压缩技术，目前在备份和归档系统中被广泛应用
- 重复数据删除的优势主要表现在两个方面：存储空间的优化和网络的优化。
- 重复数据删除可以分为两种：
  - 源端重复数据删除技术 （source de-duplication）。在数据到达目标设备之前，在源端（数据的发送端）进行重复数据删除。
  - 目标端重复数据删除技术 （target de-duplication）。在数据到达目标设备之后，在 目标端（数据的存储端）进行重复数据删除。

# 云备份 – 重复数据删除（2）

- 云备份对重复数据删除的需求：数据传输过程中网络的优化和云端存储空间的优化。
- 源端重复数据删除技术广泛应用在各种云备份环境中。
- 云备份中的源端重复数据删除技术：是指在备份数据到达云端之前，在用户端对重复备份的数据进行删除，来减少在广域网上的传输的数据量以及在云端的存储的数据量。

# 云备份 – 重复数据删除（3）

- 源端重复数据删除的效率：
  - 局部的重复数据删除（Local De-duplication）。当云备份是结合第三方提供的存储服务实现的，只能是局部的重复删除删除。
  - 全局的重复数据删除（Global de-duplication）。当云备份是以Backup SaaS的方式实现的，可以是全局的重复数据删除。